Journal of Intelligent & Robotic Systems (2020) 97:205–225
https://doi.org/10.1007/s10846-019-01057-3

Control of Multi-Agent Collaborative Fixed-Wing UASs in Unstructured Environment

A Ram Kim[1] · Shawn Keshmiri[2] · Aaron Blevins[2] · Daksh Shukla[2] · Weizheng Huang[3]

Received: 27 February 2019 / Accepted: 19 June 2019 / Published online: 5 July 2019

Abstract

In recent years, the study of dynamics and control of swarming robots and aircraft has been an active research topic. Many multi-agent collaborative control algorithms have been developed and have been validated in simulations, however the technological and logistic complexity involved in validation of these algorithms in actual flight tests has been a major hurdle impeding more frequent and wider applications. This work presents robust navigation algorithms for multi-agent fixed-wing aircraft based on an adaptive moving mesh partial differential equations controlled by the free energy heat flow equation. Guidance, navigation, and control algorithms for control of multi-agent unmanned aerial system (UASs) were validated through actual flight tests, and the robustness of these algorithms were also investigated using different aircraft platforms. The verification and validation flight tests were conducted using two different fixed-wing platforms: A DG808 sail-plane with a 4m wingspan T-tail configuration and a Skyhunter aircraft utilizing a 2.4m wingspan and a twin-boom configuration. The developed swarm navigation algorithm uses a virtual leader guidance scheme and has been implemented and optimized using optimal control theory. Multi-scale moving point guidance has been developed and complimented by a linear quadratic regulator controller. Several flight tests have been successfully conducted and a system of systems including software and hardware was successfully validated and verified.

Keywords Validation and verification · Multi-agent systems · Formation flight

1 Introduction

Swarms of unmanned aerial systems (UASs) are the inevitable future of the aerospace industry. However, without brakes or the ability for rapid deceleration, the control of large fixed-wing UASs in close proximity is one of the paramount challenges of Modern Control. Increasing the swarm size and decreasing the proximity of the agents will only amplify the complexity of collision avoidance and formation control. In addition, stability and safety concerns will grow as the number of UASs increases. Over the past two decades, large and very expensive autonomous UASs with long ranges and capability to carry large payloads have been used for Earth science and military missions. The cost and complexity of large fixed-wing UASs is many orders of magnitude higher than smaller and more mobile UASs. However, the geometry, power, and payload limitations of smaller UASs reduce their functionality and effectiveness. This problem can be addressed by distributing the tasks and payload throughout a cooperative swarm of airborne UASs. For example, the Global Hawk UAS can complete complex missions, it can carry large payloads, and is capable of conducting flights lasting up to 33 hours but each unit costs about 131.4 million dollars Ref [37]. A way to mitigate such high costs is to increase effectiveness of small but less expensive UASs by utilizing collaborative multi-agent systems distributing large payloads among agents. Recent advances in computer and computation technologies have set the stage for computation-intensive real-time UAS guidance,

✉ A Ram Kim
bellakim@iastate.edu

Shawn Keshmiri
keshmiri@ku.edu

Weizheng Huang
whuang@ku.edu

[1] Aerospace Engineering, Iowa State University, Ames, IA, USA

[2] Aerospace Engineering, The University of Kansas, Lawrence, KS, USA

[3] Department of Mathematics, The University of Kansas, Lawrence, KS, USA

navigation, and control (GNC) algorithms. Parallel computing platforms and graphical processing units (GPUs) speed up GNC computations and have the ability to push the scientific and technical boundaries of autonomous control. Advancement in microprocessor miniaturization and computation technologies has prompted a broad range of research in autonomous and collaborative robots. However, the majority of existing work are focused on small, slow, aircraft (e.g. rotary-wing UASs, quadcopters, etc.) with a relatively small payload capacity. This work represents a complete set of guidance, navigation and control (GNC) algorithms for large, fast, fixed-wing UASs capable of carrying scientific and/or military payloads (e.g. cameras, synthetic aperture radar, etc.). The effectiveness and robustness of GNC algorithms were successfully flight tested using different UAS platforms with completely different configurations (e.g. T-tail versus H-tail, different wing spans, different fuselage design).

In recent years, the study of dynamics and control of swarming robots and aircraft has been an active research topic. Many multi-agent collaborative control algorithms have been developed. The Leader-Follower scheme, the Virtual Leader-Follower, Graph theoretic methods, the Brute Force search, the Boids model, Artificial Potential Field, and Partitioning approaches are several optimization techniques used for UAS swarm and formation flights. The Leader-Follower method is probably the most popular multi-agent control method [4, 5, 16]. In this method, one of the agents is designated as the leader and the other agents are designated as followers. Simplicity is the most important advantage of this approach, however the high dependency on the leader is the main drawback. To remedy this drawback, the Virtual Leader-Follower approach was proposed where there is no physical leader among agents [4, 5, 16]. The Artificial Potential Field has been most widely used because it provides inter-collision avoidance among agents. Typically, attractive and repulsive potential fields are combined together for multi-agent systems for holding agents close via attractive forces, while avoiding collisions with each other by utilizing repulsive forces [7, 29, 42]. Repulsive potential functions (VARP) method is a reactive approach where actions are taken after position errors occur in the geometry of the desired formation shape. Another popular method for maintaining geometry of multi-agent systems is Graph Theory. A graph is a mathematical concept to visualize a group of agents as pairs using vertices and edges. Directed and undirected graph methods with different convergence rates have been used in several research approaches [18, 22, 26, 28, 31]. Both the Graph method and Virtual Structure approach assume the formation as a rigid body and no morphing in formation can be considered, which makes agents vulnerable to inter-collision and external disturbances. Optimization methods were also adopted for formation control where a cost function is defined based on the distance between agents, obstacles, and trajectories for avoiding inter-collision, obstacles, and following desired trajectories [27, 32]. More advanced topics, such as the robustness of swarming agents towards time-delays in communications or non-concurrent consensus problems of multi-agent systems with discontinuous data transmission, have been studied. However, in the majority of existing works, the dynamic constraints of fixed-wing UASs are either ignored or relaxed where GNC algorithms assume agents as 2D point masses with no aerodynamic effects [3, 17, 19, 21, 33]. Fixed-wing UASs have several state constraints (including stall speed, stall angle of attack, and structural loading) that are not considered in these optimization techniques. Simple linear controllers that lack robustness and adaptation (e.g. Proportional-Integral-Derivative) are used for inner-loop control, and guidance and navigation logics have been designed for slow moving robots or quadcopters [3, 17, 19, 21, 33]. The nonlinear and unsteady nature of swarming problems creates a demand for advanced controllers. Nonlinear model predictive control (NMPC) and adaptive fuzzy control have been used for control of autonomous vehicles in the presence of control and state constraints [8, 24, 36]. Ref. [24, 36] prove that all closed-loop control signals are semi-globally uniformly ultimately bounded (SGUUB) while full state constraints are satisfied. This is particularly very important for collaborative flight tests where UASs fly in close proximity of each other, where any violation of state or control constraints can easily cause catastrophic accidents. Ref. [8] shows control constraints are satisfied, however states are not constrained as it is shown in Ref. [24, 36].

The point mass assumption is acceptable for small and slow robots; however, such assumptions cannot be made on large UASs where spatiotemporal requirements necessitate close formations. In navigation algorithms, the state of consensus between agents is used for aggregation of robots [26, 27]. However, Earth science missions demand path planning and UASs flight formation. Spatial tracking of multi-agent aircraft transforms guidance and navigation problems into a high-dimensional space problem and the curse of dimensionality makes convergence of optimization in real-time challenging. The stability of guidance logic is essential for the long term coherency of swarming UASs. Challenging subjects such as morphing, splitting, and algorithm scalability are either neglected or investigated in structured environments with no external disturbances such as wind, gust, and windsheer. Most scalability and shape optimization algorithms for swarms of multi-agent systems are only designed to support computer-based animations with no references to fixed-wing UASs with high speeds and high inertia [1, 2]. Probably the most relevant research was conducted by the Vijay Kumar Lab at the University

Fig. 1 DG-808 and Skyhunter UAS

of Pennsylvania, where demonstrations included search, tracking, and classification of targets with small multi-agent rotary-wing UASs that can "Stop, Think, and Proceed" [19]. Literature review shows that most UAS swarm simulations are started from optimal initial conditions (e.g. trim with no rotational acceleration nor sideslip). However in actual flight tests, achieving trim conditions is almost impossible [1, 3–7, 9, 10, 16–22, 26–29, 31–33, 38, 41, 42]. Practically, the initial conditions of the aircraft at the time of transition to automatic control are unpredictable and the asynchronous engagement of aircraft autopilot systems only exacerbate the complexity of initializing swarms of UASs in actual flight tests. Such randomness in the initial conditions and the inability of large UASs to brake or rapidly decelerate can quickly result in collision amongst the agents, or instability in swarm control algorithms. This work presents implementation and integration of adaptive moving mesh partial differential equations and validation and verification of guidance, navigation and control algorithms using different UAS platforms in actual flight test experiments.

2 UAS 6-DoF Dynamic Model

This section describes the dynamic models used in this research work. Dynamic modeling is an essential part of GNC algorithms since the behavior of the systems will be observed for different parameter designs based on various aircraft platforms. Additionally, six degress of freedom (6 DoF) nonlinear equations of motions are construed in this work not to assume the platform as a point mass. In order to develop GNC algorithms for real world application, the 6 DoF model is more appropriate than that of a point mass approach. Lastly, the dynamic model is essential for accurately propagating aircraft states in the simulation environment in Simulink and the hardware-in-the-loop (HiTL) testbed.

Two distinct platforms have been used in this work: DG808 and Skyhunter UAS. The DG808 is the glider type UAS which has the long wingspan (13.08 ft). The Skyhunter has the shorter wingspan of about 7 ft, though both platforms have a similar weight of around 9 lbs. The DG808 has a faster trim

Table 1 DG808 and Skyhunter UAS characteristics [H]

Parameters	Unit	DG808	Skyhunter
S	ft^2	6.96	4.82
$\bar{c}$	ft	0.6	0.731
b	ft	13.08	6.875
AR	$\sim$	25.89	9.81
V_{trim}	ft/s	59.07	50.6
V_{trim}	$knots$	35	30
$Mass$	lbm	9.53	8.27
$I_{xx_{Body}}$	$slug-ft^2$	1.1	0.44
$I_{yy_{Body}}$	$slug-ft^2$	0.3	0.54
$I_{zz_{Body}}$	$slug-ft^2$	1.43	0.35
$I_{xz_{Body}}$	$slug-ft^2$	0	0

speed (35 knots) than the Skyhunter has (30 knots). Different avionics payloads were used for each platform and are presented in detail at Section 6.1. Figure 1 presents both platforms. The geometric characteristics are presented in Table 1.

To build the 6 DoF nonlinear equations of motion (EoM), Newton's second law (translational motion) and Euler's law (rotational motion) have been used Ref. [35]. The inertial and body frames have been defined to consider the forces and moments acting on the rigid body and to incorporate sensor data appropriately. The inertial frame ($\hat{I}$) is fixed at the Earth and has three orthonormal axes: north, east, down. The body frame ($\hat{B}$) is fixed at the center of gravity (c.g.) of the aircraft and defined along three orthonormal axes: x_B, y_B, and z_B. Equations 1 to 16 represents 16 differential equations as the mathematical representation of the aircraft motion including 6 DoF nonlinear EoM (Eqs. 1 to 6). The kinematic equations are needed to generate aircraft attitude angles (Eqs. 7 to 9) since there are no physical sensors in the simulation environment. Equation 5 through Eq. 13 presents the servo dynamics. Eqution 14 through Eq. 16 describes the position coordinated in the inertial frame.

$$\dot{U} = RV - QW - g\sin\theta + (X_T + X_A)/m \tag{1}$$

$$\dot{V} = -RU + PW + g\sin\phi\cos\theta + Y_A/m \tag{2}$$

$$\dot{W} = QU - PV + g\cos\phi\cos\theta + Z_A/m \tag{3}$$

$$\begin{aligned}\dot{P} = {} & J_{XZ}[J_X - J_Y + J_Z]PQ \\ & -[J_Z(J_Z - J_Y) + J_{XZ}^2]QR + J_Z L_A \\ & +J_{XZ}N_A/(J_X J_Z - J_{XZ}^2)\end{aligned} \tag{4}$$

$$\begin{aligned}\dot{Q} = {} & ([J_Z - J_X]PR - J_{XZ}[P^2 - R^2] + M_A + M_T) \\ & /(J_X J_Z - J_{XZ}^2)\end{aligned} \tag{5}$$

$$\begin{aligned}\dot{R} = {} & ([(J_X - J_Y)J_X + J_{XZ}^2]PQ \\ & -J_{XZ}[J_X - J_Y + J_Z]QR \\ & +J_{XZ}L_A + J_X N_A)/(J_X J_Z - J_{XZ}^2)\end{aligned} \tag{6}$$

$$\dot{\phi} = P + \tan\theta(Q\sin\phi + R\cos\phi) \tag{7}$$

$$\dot{\theta} = Q\cos\phi - R\sin\phi \tag{8}$$

$$\dot{\psi} = (Q\sin\phi + R\cos\phi)/\cos\theta \tag{9}$$

$$\dot{\delta_T} = \delta_T/\tau_T + \delta_T^{cmd}/\tau_T \tag{10}$$

$$\dot{\delta_E} = \delta_E/\tau_E + \delta_E^{cmd}/\tau_E \tag{11}$$

$$\dot{\delta_A} = \delta_A/\tau_A + \delta_A^{cmd}/\tau_A \tag{12}$$

$$\dot{\delta_R} = \delta_R/\tau_R + \delta_R^{cmd}/\tau_R \tag{13}$$

$$\begin{aligned}\dot{p_N} = {} & U\cos\theta\cos\phi + V(-\cos\phi\sin\psi \\ & +\sin\phi\sin\theta\cos\psi) \\ & +W(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi) - w_N\end{aligned} \tag{14}$$

$$\begin{aligned}\dot{p_E} = {} & U\cos\theta\sin\phi + V(\cos\phi\cos\psi \\ & +\sin\phi\sin\theta\sin\psi) \\ & +W(-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi) - w_E\end{aligned} \tag{15}$$

$$\dot{p_H} = U\sin\theta - V\sin\phi\cos\theta - W\cos\phi\cos\theta - w_H \tag{16}$$

{U,V,W} are the translational velocity components aligned with $\hat{B}$. {P,Q,R} are the rotational velocities (roll, pitch, and yaw rate, respectively) in $\hat{B}$. $\{\phi, \theta, \psi\}$ are the Euler angles (roll, pitch, yaw angle, respectively) in $\hat{I}$. $\{p_N, p_E, p_H\}$ are the position coordinates (north, east, and height) in $\hat{I}$. $\{L_A, M_A, N_A\}$ represents the aerodynamic moments in $\hat{B}$. Since the motors are installed above the center of gravity in z_B direction for both platforms, only thrust pitching moment exists (M_T). $\{X_A, Y_A, Z_A\}$ describes the aerodynamic force in $\hat{B}$. $\{X_T\}$ is the thrust force from the motor. The aerodynamic forces and moments are estimated with the component build-up method, see Ref. [30]. $\{J_X, J_Y, J_Z\}$ indicates the moment of inertia in $\hat{B}$ and J_{XZ} is the moment of inertia in the x_B and z_B direction. m is the aircraft mass. Due to the availability of the sensors onboard, the airspeed of the aircraft (V_T) is measured by the pitot tube. Airflow angles $\{\alpha, \beta\}$ are estimated by an Extended Kalaman filter. Then, the body velocity can be found by the following equations: $U = V_T\cos\alpha\cos\beta$, $V = V_T\cos\beta$, and $W = V_T\sin\alpha\sin\beta$, see Ref. [35]. $\{\tau_T, \tau_E, \tau_A, \tau_R\}$ are the control surface deflections of the throttle, elevator, aileron, and rudder, respectively. $\{W_N, W_E, W_H\}$ are the wind velocity in north, east, and height directions.

In order to extract the state space model for the controller design, the 6 DoF nonlinear EoM should be linearized and perturbed. The perturbation value is defined as the difference between the total state (x) and the trim state (x_{trim}) : $\Delta x(t) = x(t) - x_{trim}$. The states are defined as follows: $x^T = [V_T, \alpha, \beta, \phi, \theta, \psi, P, Q, R, \delta_t, \delta_e, \delta_a, \delta_r, p_N, p_E, p_H]$. Trim conditions of level-wing recti-linear flight for each platform are defined in Table 2.

The LTI dynamic models were determined as follows for DG808 and Skyhunter.

Longitudinal, DG808:

$$\begin{bmatrix}\dot{u}\\ \dot{\alpha}\\ \dot{\theta}\\ \dot{q}\end{bmatrix} = \begin{bmatrix}-0.2030 & -7.7004 & -32.1874 & 0\\ -0.0177 & -9.6416 & -0.0151 & 0.9632\\ 0 & 0 & 0 & 1\\ 0.2003 & -121.3565 & 0.0093 & -5.4812\end{bmatrix}\begin{bmatrix}u\\ \alpha\\ \theta\\ q\end{bmatrix} + \begin{bmatrix}3.0391 & -0.8475\\ 0 & -0.3067\\ 0 & 0\\ -1.7311 & -40.8156\end{bmatrix}\begin{bmatrix}\delta_t\\ \delta_e\end{bmatrix} \tag{17}$$

Table 2 Trim conditions for DG808 and Skyhunter

States/control	DG808	Skyhunter
α_{trim}	1.5 deg	0.7 deg
$V_{T_{trim}}$	35 knots	30 knots
$\delta_{E_{trim}}$	−1.7 deg	8.27 deg

Lateral, DG808:

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -0.4412 & 0.5449 & -0.0074 & -0.9847 \\ 0 & 0 & 1 & 0 \\ -10.7386 & 0 & -20.9688 & 2.9327 \\ 10.0800 & 0 & -1.2881 & -0.6602 \end{bmatrix} \begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0.1616 \\ 0 & 0 \\ 140.5722 & 0.6242 \\ -6.7747 & -6.4997 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \tag{18}$$

Longitudinal, Skyhunter:

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -0.1240 & 19.0662 & -32.1974 & 0 \\ -0.0250 & -6.2646 & -0.0080 & 0.9405 \\ 0 & 0 & 0 & 1 \\ 0.0224 & -14.6920 & 0.0051 & -2.7085 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 5.9203 & -0.7755 \\ -0.0544 & -0.3158 \\ 0 & 0 \\ 0.2737 & -19.4782 \end{bmatrix} \begin{bmatrix} \delta_t \\ \delta_e \end{bmatrix} \tag{19}$$

Lateral, Skyhunter:

$$\begin{bmatrix} \dot{\beta} \\ \dot{\phi} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -0.6048 & 0.6359 & -0.0153 & -0.9797 \\ 0 & 0 & 1 & 0 \\ -35.6645 & 0 & -8.2161 & 2.4729 \\ 34.3765 & 0 & -1.3188 & -2.1512 \end{bmatrix} \begin{bmatrix} \beta \\ \phi \\ p \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0.2484 \\ 0 & 0 \\ 74.0763 & 3.8384 \\ 0.2236 & -26.3674 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \tag{20}$$

Table 3 presents the modal analysis for both platforms. Skyhunter has the slower short period mode compared to the DG808. Additionally, the damping ratio is very low in the phugoid mode for both platforms. Due to the longer wingspan of the DG808, the roll mode is slower than the Skyhunter's roll mode. The spiral of both aircrafts are unstable. The low damping ratio in the phugoid mode and the unstable spiral mode cause challenges in the control tasks.

Table 3 DG808 and Skyhunter modal analysis

Mode	Unit	DG808	Skyhunter
Longitudinal			
ω_{SP}	rad/s	13.01	5.63
ζ_{SP}	–	0.582	0.81
τ_{SP}	s	0.132	0.22
$\omega_{phugoid}$	rad/s	0.879	0.6691
$\zeta_{phugoid}$	–	0.094	0.013
$\tau_{phugoid}$	s	12.1	106
Lateral			
ω_{DR}	rad/s	3.38	6.3024
ζ_{DR}	–	0.19	0.174
τ_{DR}	s	1.56	0.89
τ_{Spiral}	s	−19.4	−68.1
τ_{Roll}	s	0.048	0.01114

3 The Navigation for the Virtual Leader and Multi-Scale Moving Point Guidance

This section discusses the foundation of the navigation and guidance algorithms for the multi-agent systems. In order to perform multi-agent flight, the guidance algorithm has been decentralized for each agent and the virtual leader scheme has been implemented to enhance the robustness of the single-point failure. The single point failure is the most common reason for errors in the leader-follower schemes since the mission cannot be completed if the leader becomes incapable of operating missions by any reason (e.g. communication loss, stall etc.). Basically, followers depend on the leader's states (e.g. position, attitudes etc.) in the leader-follower scheme. In this work, the virtual leader has been used and provides the reference of position and velocity for all agents. Therefore, to accomplish the mission, the navigation for the virtual leader should be developed. In this work, the mission is to follow the race track pattern in the flight test field. The LQ guidance algorithm has been modified to be used as the navigation algorithm for the virtual leader. Ref. [25] presents the LQ guidance algorithm and provides a desired lateral acceleration command and converts it into a heading angle change command in order to track the desired trajectory. It is imperative for the guidance portion of the autonomous system to compute the appropriate attitude angle commands (i.e. the roll and

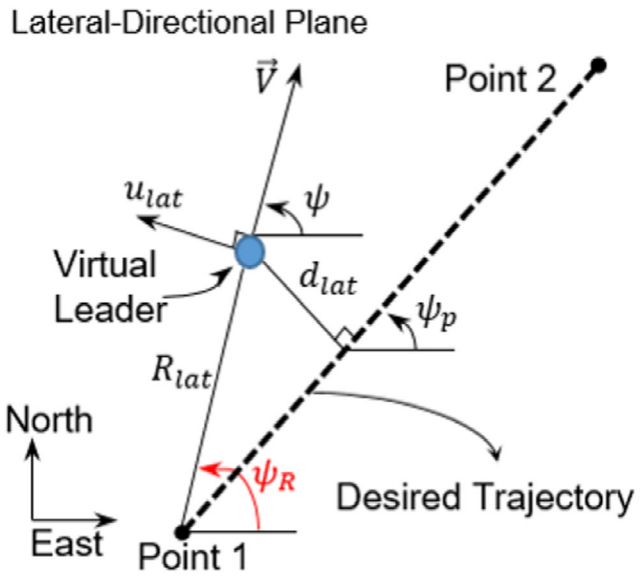

Fig. 2 LQ guidance geometric description

pitch angles) required for the aircraft to follow the desired trajectory generated by the navigation algorithms.

3.1 LQ Path Planning for the Virtual Leader

At the beginning of the navigation algorithm, LQ path planning updates the virtual leader's position for the next sample time (k). The dynamics of the cross-track error (d_{lat} in Fig. 2) has been developed to find the optimal lateral acceleration as follows:

$$\dot{d}_{lat} = |\mathbf{V}| \sin(\psi - \psi_p) \tag{21}$$

where $|\mathbf{V}|$ is the desired speed; ψ is the heading angle (in this diagram, it was defined from the east axis); ψ_p is the heading angle of the desired trajectory.

In order to apply LQR control methods, the cross-track error(d_{lat}) is considered as a bounded or maximum allowable error (d_b). That is, we assume that the cross-track error is always less than the maximum allowable error ($d_{lat} \leq d_b$). Therefore, the lateral acceleration (u_{lat}) is presented as follows:

$$u_{lat} = |\mathbf{V}|\dot{\psi} \tag{22}$$

Next, the LTI (linear time invariant) state space is formulated as follows:

$$\begin{bmatrix} \dot{d}_{lat} \\ \dot{v}_{d_{lat}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d_{lat} \\ v_{d_{lat}} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_{lat} \tag{23}$$

By using the quadratic cost function (Eq. 24) and solving the Ricatti equation, the lateral acceleration command (Eq. 25) is derived as follows:

$$\mathbf{J} = \frac{1}{2}\int_{\infty}^{t_0} \mathbf{x}^T \mathbf{Q}\mathbf{x} + Ru^2(t)dt, \quad \mathbf{Q} = \begin{bmatrix} q_1^2 & 0 \\ 0 & q_2^2 \end{bmatrix} \tag{24}$$

$$u_{lat} = -[q_1 d_{lat} + \sqrt{2q_1 + q_2^2}\|v\|], \quad q_1 = |\frac{d_b}{d_b - d_{lat}}| \tag{25}$$

where $\mathbf{x}$ and u are the states and control vector, respectively; $\mathbf{Q}$ and R are the weighting matrices for states and control, respectively. R is assumed to be 1. While the Ricatti equation is solved, q_1 is expressed as a function of the maximum allowable error (d_b) and the current cross-track error (d_{lat}).

Finally, the heading angle rate ($\dot{\psi}$) can be determined via Eq. 22, where aircraft dynamic constraints have been applied in order to plan physically feasible paths for the fixed-wing vehicle using the maximum allowable roll angle for the aircraft (see Eq. 26).

$$\Delta\psi \leq \frac{g}{V_T} \sin\phi_{max} \cdot \Delta t \tag{26}$$

3.2 Multi-Scale Moving Point Guidance

In order to describe the guidance algorithms used in this work, the moving point should first be introduced. The moving point is mathematically defined by the coordinates

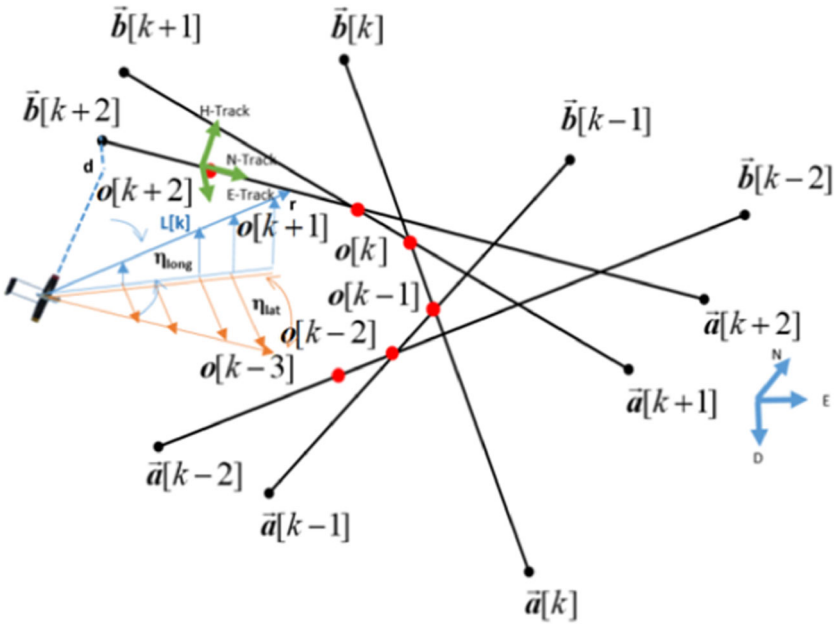

Fig. 3 Waypoint generation logic and distance and angle errors

of position and velocity in the inertial coordinate system (Eq. 27).

$$o[k] = \begin{cases} \mathbf{p}^o[k] = p_N^o[k],\, p_E^o[k],\, p_H^o[k] \\ \mathbf{v}^o[k] = v_N^o[k],\, v_E^o[k],\, v_H^o[k] \end{cases} \tag{27}$$

The biggest advantage of using moving points is that the desired trajectory can be treated as an arbitrary shape by forming a line segment ($\overrightarrow{a}[k]$ and $\overrightarrow{b}[k]$) as the desired trajectory at each time step using two consecutive moving points ($o[k]$ and $o[k-1]$), see Fig. 3. For example, this feature has been complimented with the morphing potential field algorithm for collision avoidance, since the classical navigation algorithm could not handle a curved path.

Ref. [23] investigated line-of-sight trajectory following in the lateral plane. Then, Ref.[8] has expanded to 3D space. The multi-scale moving point guidance algorithm is a variation of guidance from Ref. [8, 23]. To articulate the modification, some geometric characteristics are presented in Fig. 4, where $\overrightarrow{d}$ is the projected aircraft position on the line segment consisting of $\overrightarrow{A}$ and $\overrightarrow{B}$. Using a predefined length ($d_{dR_{Lat}}$), the reference point on the lateral plane ($\overrightarrow{R}_{Lat}$) can be found using Eq. 28.

$$\overrightarrow{R}_{Lat} = \overrightarrow{d} + \frac{\overrightarrow{B} - \overrightarrow{d}}{\| \overrightarrow{B} \| - \| \overrightarrow{d} \|} d_{dr_{Lat}} \tag{28}$$

A similar guidance approach is used in the longitudinal plane. Modifications from previous work involves unique positioning of the reference point in the lateral and longitudinal planes in order to reflect their respective aircraft dynamic modes. The predefined length in the lateral plane ($d_{dR_{Lat}}$) is different than the one in the longitudinal plane ($d_{dR_{Lon}}$) so that the reference points are separated. These points are also aligned with the dynamic modes of the fixed-wing aircraft, where the roll and short period modes vary significantly. The positioning of these reference points have a significant effect on the calculation of error angles (η_{Lat} and η_{Lon}) as follows:

$$\eta_{Lat} = \tan^{-1} \frac{[r_N - p_N]_T}{[r_E - p_E]_T} - \tan^{-1} \frac{V_{track_N}}{V_{track_E}} \tag{29}$$

$$\eta_{Lon} = \tan^{-1} \left(\frac{V_{track_H}}{V_{track_{NE}}} \right) - \tan^{-1} \left(\frac{L_{track_H}}{L_{track_{NE}}} \right) \tag{30}$$

The basic principle for the multi-scale guidance is to convert the error angles to the attitude commands independently in the lateral and longitudinal plane. Once the error angles are computed, the roll and pitch angle commands can be determined. The pitch angle command is calculated using an approach from Ref. [8]:

$$\theta_{cmd} = \tan^{-1} \left\{ k_{a_{Lon}} \left(k_{p_{Lon}} \eta_{Lon} + k_{i_{Lon}} \int \eta_{Lon} dt \right) \right\} \tag{31}$$

where k_p and k_i are the proportional and integral gains. For the aircraft roll angle command, the lateral acceleration is converted to ϕ_{cmd} as presented in Ref. [12]:

$$a_{cmd} = \frac{V_T^2}{R} \quad \text{where} \quad R = \frac{L_{Lat}}{2 \sin \eta_{Lat}} \tag{32}$$

where R is the resultant desired turning radius shown in Fig. 4. L_{Lat} is defined as the distance between the aircraft

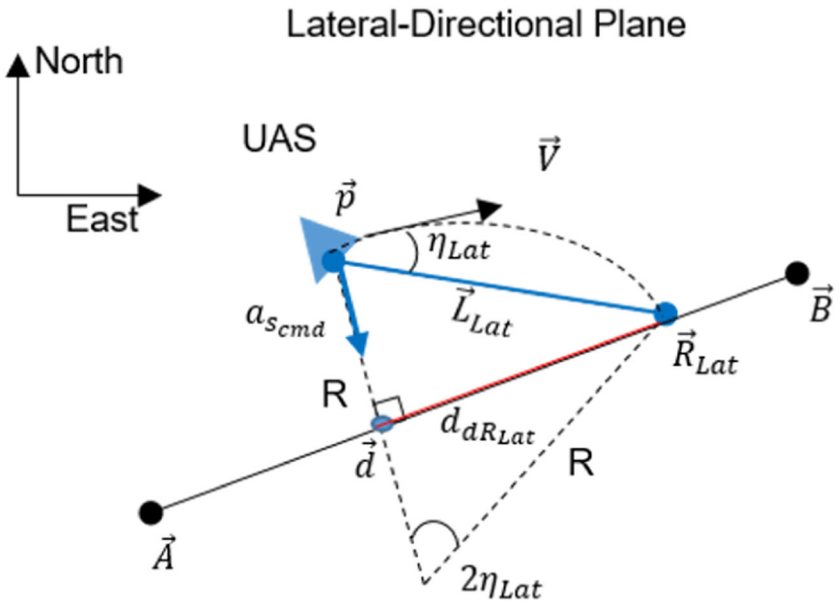

Fig. 4 Lateral guidance geometric description

position and the reference point ($\mathbf{R}_{lat}$). The roll angle command is calculated using the following equation:

$$\phi_{cmd} = \tan^{-1} \frac{a_{cmd}}{g} \tag{33}$$

Finally, there are two remaining guidance commands: the airspeed ($V_{T_{cmd}}$) and the sideslip angle (β_{cmd}). The sideslip angle command (β_{cmd}) is always designated to zero for coordinated turning. The airspeed command ($V_{T_{cmd}}$) is assigned as the current aircraft airspeed when the autopilot is engaged in order to prevent abrupt behavior at the beginning of autonomous flight.

In summary, the guidance algorithm provides four state commands: $V_{T_{cmd}}$, θ_{cmd}, ϕ_{cmd} and β_{cmd}. These commands have are used as the reference values for the LQR controller in order to follow the desired trajectory.

4 Navigation of Multi-Agent UASs

The moving mesh method is utilized for path planning of multi-agent systems by solving the moving mesh partial differential equations (PDEs), Ref. [15]. The adaptive properties of the moving mesh method is derived by the free energy heat flow equation. As observed in the flocking of birds, the moving mesh PDE is formulated according to the M-uniformity requirement so the mesh points are distributed uniformly as specified by a metric (M). Using the evolutionary physical features of swarming birds, where the distances are normalized as a function of the proximity of other birds, corrective actions in the swarm are calculated based on principles in the moving mesh method. This method is specifically designed for the numerical solution of PDEs. The position of any agent is denoted by $\mathbf{p^o} = [\mathbf{p_N^o}, \mathbf{p_E^o}, \mathbf{p_H^o}]^T$, where the superscript T stands for the transpose of a vector or a matrix. From Ref. [11], the mesh velocity is given as:

$$\frac{d\mathbf{p}^o}{dt} = \frac{1}{\tau} \sum_{K \in w_o} |K| \mathbf{v_i}_o{}^K \tag{34}$$

In the above equation, τ is a positive and constant parameter used to adjust the time scale of mesh motion; w_o is the collection of the mesh elements for vertice i; $|K|$ is the volume of element K; and i_o and v_i are the local index and local velocity of K, respectively. The spatial dimension is denoted by d (d = 2 for 2D and d = 3 for 3D). Finally, the local velocities for the d + 1 vertices of element K (with their position denoted by $\mathbf{p_0}^K$, ..., $\mathbf{p_d}^K$) are given as:

$$\begin{bmatrix} (\mathbf{v_1^k})^T \\ \vdots \\ (\mathbf{v_d^k})^T \end{bmatrix} = -GE_K^{-1} + E_K^{-1} \frac{\partial G}{\partial J} \hat{E} E_K^{-1} + \frac{\partial G}{\partial r} \frac{det(\hat{E})}{det(E_K)} E_K^{-1} \tag{35}$$

$$\mathbf{v_o^K} = -\left(\mathbf{v_1^K} + \ldots + \mathbf{v_d^K}\right) \tag{36}$$

In the above equation, "det" denotes the determinant of a matrix, E_K is the edge matrix of size $d \times d$ and is defined as $E_K = [\mathbf{P}_1^K - \mathbf{P}_0^K, \cdots, \mathbf{P}_d^K - \mathbf{P}_0^K]$, and $\hat{E}$ is the edge matrix of the unitary equilateral simplex in d dimensions. The quantities G are associated with the objective function, which is minimized for an optimal mesh. To make the mesh as uniform as possible, we will use an objective function, based upon the so-called equidistribution and alignment conditions (Ref. 2). The quantities are given as:

$$G = \frac{1}{3}\left(trace\left(\hat{E}E_K^{-1}E_K^{-T}\hat{E}^T\right)\right)^d + \frac{1}{3}d^d det\left(\hat{E}E_K^{-1}\right)^2 \tag{37}$$

$$\frac{\partial G}{\partial J} = \frac{2d}{3}\left(trace\left(\hat{E}E_K^{-1}E_K^{-T}\hat{E}^T\right)\right)^{d-1} E_K^{-T}\hat{E}^T \tag{38}$$

$$\frac{\partial G}{\partial r} = \frac{2}{3}d^d det\left(\hat{E}E_K^{-1}\right) \tag{39}$$

Notice that the first and third quantities are scalar while the second one is a matrix of size d×d. The moving mesh algorithm is robust against ***the morphing and splitting*** formation shape since the objective function of equidistribution and alignment (Eq. 35–39) will automatically correct the inner agents' desired position corresponding to the updated formation shape or a boundary condition. Additionally, the moving mesh methods have been developed for any number of mesh points in any dimensional space, meaning it ***accommodates the scalability***. UASs' receding horizon moving points (see Fig. 5) have fixed distances from the virtual leader in the North-East-Down coordinate system. Agents will be positioned using the desired relative distance and the heading angle with respect to the virtual leader.

As the virtual leader moves in real-time, the outer agents will be re-positioned in order to maintain formation coherency. The following equations depict the position update for the outer agents in this logic.

$$p_{N_i} = p_{N_{RP}} - \|E\| \cdot \sin\psi_{RP} + \|N\| \cdot \cos\psi_{RP} \tag{40}$$

$$p_{E_i} = p_{E_{RP}} + \|E\| \cdot \cos\psi_{RP} + \|N\| \cdot \cos\psi_{RP} \tag{41}$$

Above, p_{N_i} is the north position of the i^{th} agent and $p_{N_{RP}}$ is the north position of the virtual leader. $\|E\|$ is a desired relative distance for the east direction, and $\|N\|$ is a desired relative distance for the north direction. ψ_{RP} is the heading angle with respect to the virtual leader.

5 UAS LQR Optimal Flight Control

In this work, Linear Quadratic Regulator (LQR) controllers have been designed for two platforms: DG808 and Skyhunter. The guidance algorithm generates four desired states: $V_{T_{cmd}}, \phi_{cmd}, \theta_{cmd}$ and β_{cmd}. They are utilized as the reference signal for the nonzero set point. While the nature of LQR aims to regulate all states to zero, nonzero set points have been used to follow the desired trajectory. The LTI

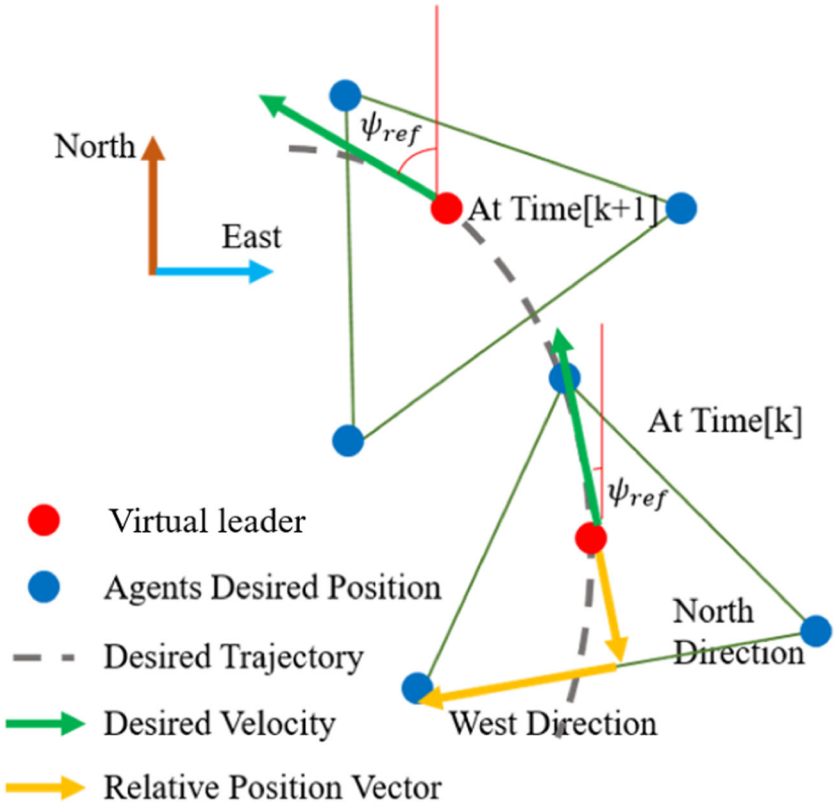

Fig. 5 Virtual leader formation algorithm geometric description

state space has been developed previously in Section 2. The augmented LTI states and control vectors with guidance commands have been defined to implement the nonzero set point as follows:

$$\mathbf{x}_{long} = \begin{bmatrix} u \\ \alpha \\ \theta \\ q \\ \int u_{err} \\ \int \theta_{err} \end{bmatrix}, \quad \mathbf{u}_{long} = \begin{bmatrix} \delta_T \\ \delta_E \end{bmatrix} \tag{42}$$

$$\mathbf{x}_{lat} = \begin{bmatrix} \beta \\ \phi \\ p \\ r \\ \int \beta_{err} \\ \int \phi_{err} \end{bmatrix}, \quad \mathbf{u}_{lat} = \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \tag{43}$$

The quadratic cost function for the LQR controller is presented as follows:

$$\mathbf{J} = \frac{1}{2} \int_{\infty}^{t_0} \mathbf{x}^T \mathbf{Q}\mathbf{x} + \mathbf{u}^T \mathbf{R}\mathbf{u} \quad dt \tag{44}$$

The Q and R weighting matrices are designed to maximize the tracking quality while ensuring smooth control surface maneuvers.

6 Validation and Verification

6.1 Software and Hardware in the Loop Simulations

Prior to flight testing, verification of the aircraft software and hardware systems is imperative to mitigate any potential abnormal behavior of the system. For this purpose, software-in-the-loop (SiTL) and hardware-in-the-loop (HiTL) test-beds have been developed and are thoroughly documented in Ref. [13]. The purpose of SiTL testing is to ensure that the software has been correctly implemented onto the aircraft avionics. The purpose of HiTL is to ensure that the hardware and software are properly working together, and this testing is done using an identical setting to that of flight testing. In this test, the communication links, sensor integration, and the behavior of control surface etc. can be observed in real time and tested with the ground station (Fig. 6). Additionally, the nonlinear 6 DoF EoM have been implemented onto the flight computer in order to generate the simulation states and test the corresponding control outputs to the aircraft via servo commands.

The DG808 and Skyhunter platforms have different avionics configurations. For the DG808 aircraft, an Nvidia TK1 (four high-performance ARM cores and a 256 core GPU) is used as the flight computer with a customized Data Acquisition board (DAQ) to handle PWM signals. A

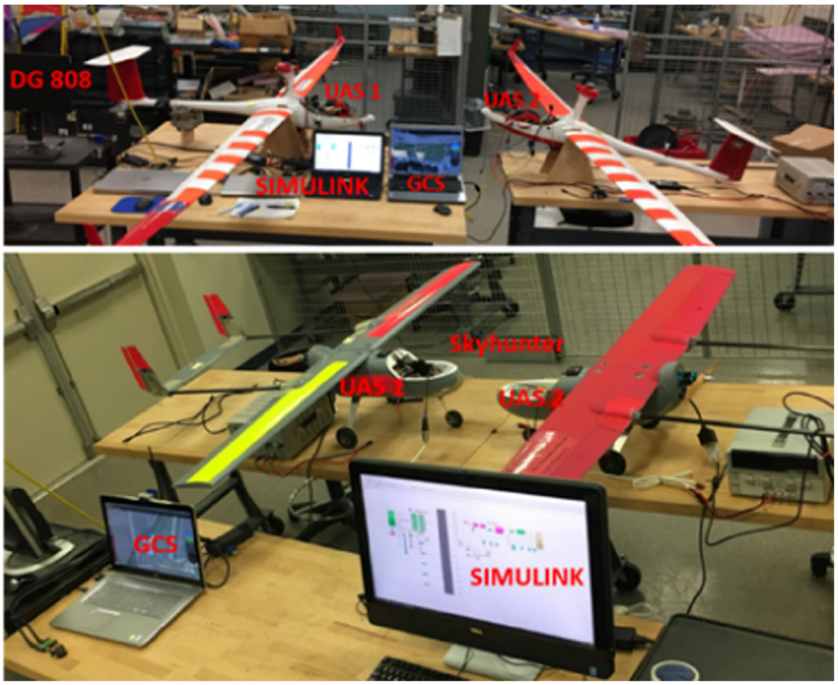

Fig. 6 HiTL Simulation Setup for DG808 and Skyhunter

VectorNav 200 is used as the IMU sensor. The Skyhunter platforms utilize an Odroid XU4 as the flight computer and a Pixhawk unit as an IMU sensor package.

Fig. 7 Swarm of two DG-808 and Skyhunter UAS in autonomous flight

6.2 Validation and Verification Flight Tests

In order to verify and validate the essential parts of the developed GNC algorithms (multi-scale guidance logic, collision avoidance algorithms, and the LQR controller) beyond the standard software and hardware in the loop simulations, two flight tests were conducted in unstructured environments and flight test results are presented in this section. Structured environments are highly controlled indoor test facilities. In structured environments, aircraft experience no or little external disturbances (e.g. wind, gust, etc.). In structured environments, space limitations dictate size and speed of UASs used in experiments (e.g. usually small and slow robots). Unstructured environments are complex and have an inherent uncertainty mainly due to external disturbances (e.g. wind, gust, etc.). There is no a priori information on how and when external disturbances change their amplitudes and/or directions. Such uncertainties increase dimensional state space of control problems. In unstructured environments, speed and size of UASs are constrained by the aircraft performance and design criterion and not by the size of the test facility.

Two different UAS platforms were used for validation flight tests, one was held on April 2017 using two of DG808 UASs (Fig. 7), Ref. [39, 40]. The other one was held on April 26th, 2018 using two of the Skyhunter UASs.

The scenario of both flight tests is presented below:

– Two agents have to follow a race track pattern due to the confined flight test area.
– One of the agents will be ahead of the other agent.

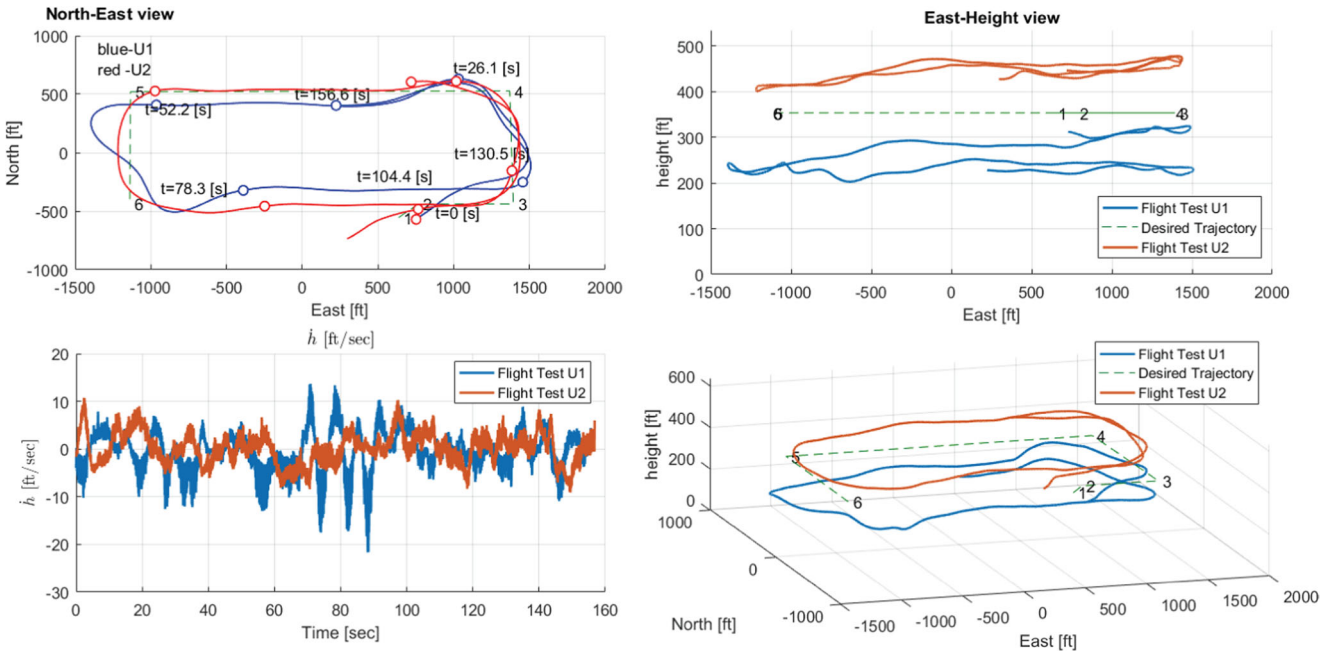

Fig. 8 Flight Test 1: 1st autopilot engagement Position Tracking DG808

– Pilots engaged the autopilot mode while each aircraft is in a steady-level trim condition.

For the DG808 flight tests, autopilots were engaged at **different initial conditions** in order to demonstrate the robustness of GNC algorithms toward random initial conditions. The positional tracking, longitudinal and lateral directional states and controls are presented. The average wind speed during this flight test was 4.6 mph.

Figure 8 presents the position tracking in the longitudinal (top left) and lateral (top right) plane for DG808 UASs' first swarm engagement flight test. Additionally, the rate of altitude change ($\dot{h}$, bottom left) and 3D view (bottom right) have been presented. Both agents engaged the autopilot near the southwest corner of the race track and followed the pattern in the counterclockwise direction. Waypoints 3 to 6 have been uploaded from the ground station, while waypoint 1 (the average position of two agents during

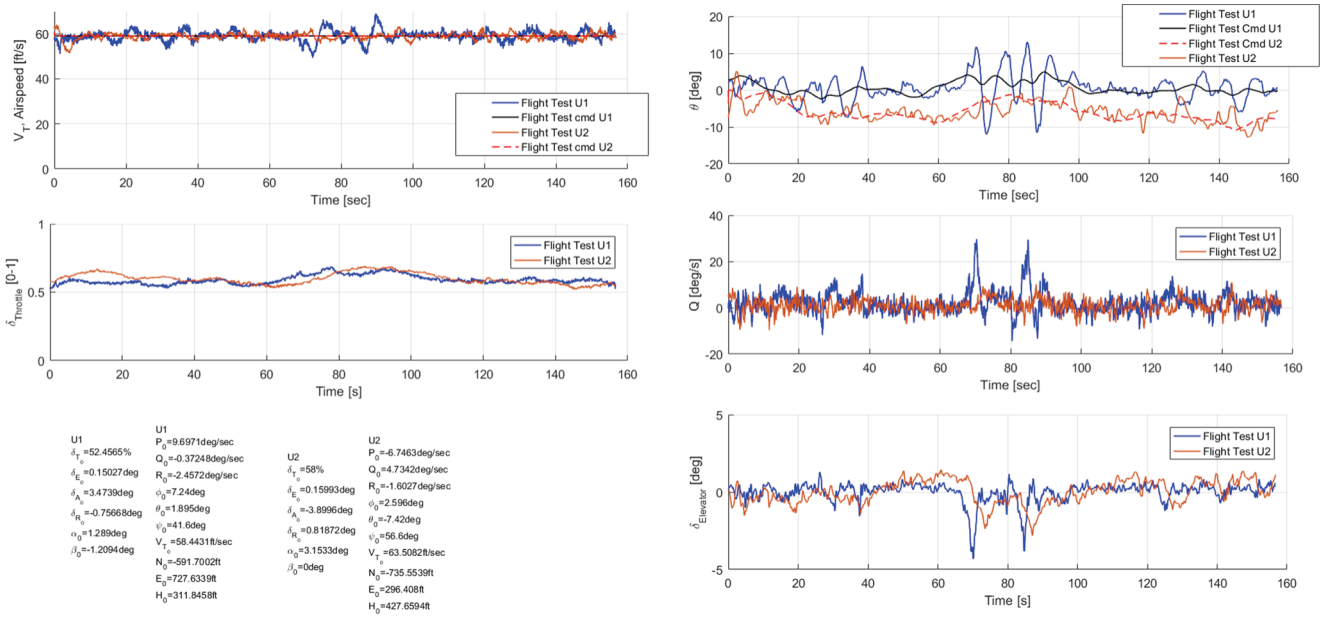

Fig. 9 Flight Test 1: 1st autopilot engagement DG808 Longitudinal States

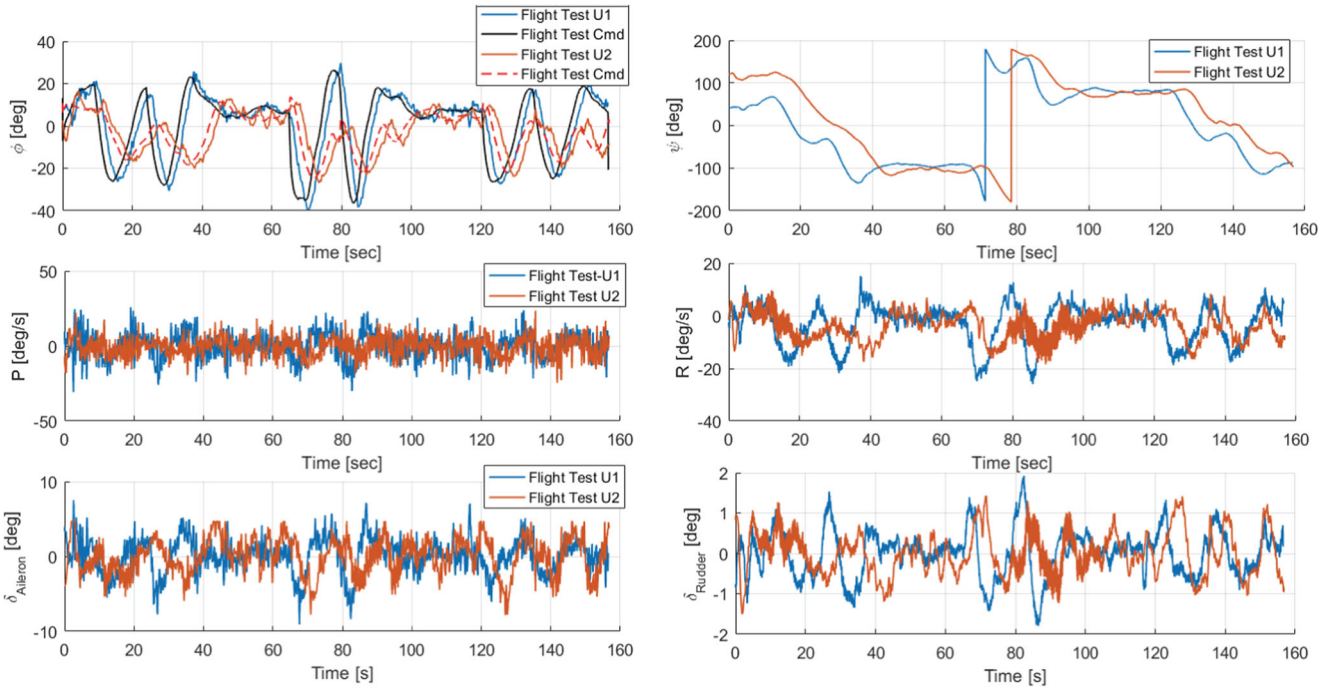

Fig. 10 Flight Test 1: 1st autopilot engagement DG808 Lateral-Directional States

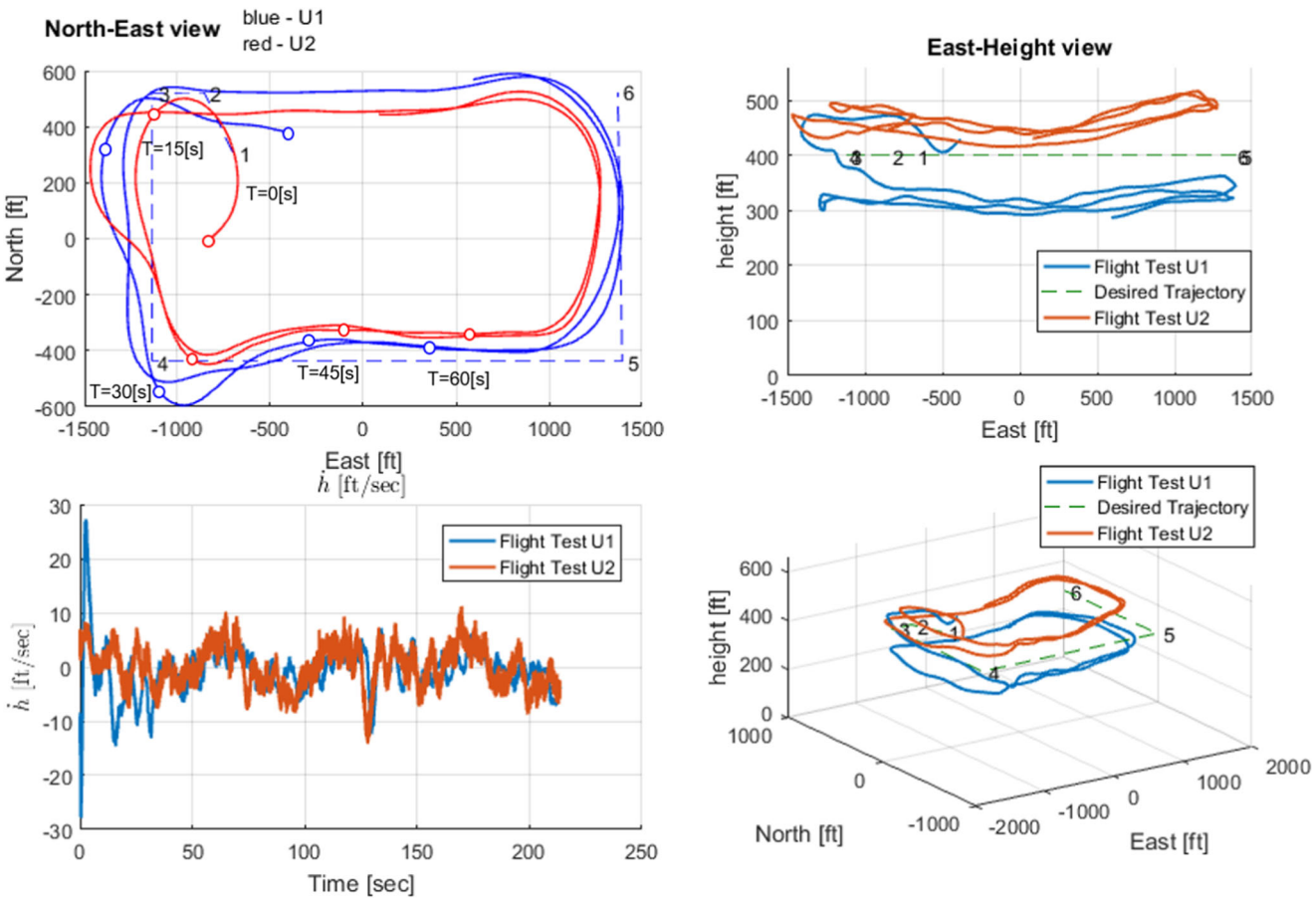

Fig. 11 Flight Test 1: 2nd autopilot engagement position tracking DG808

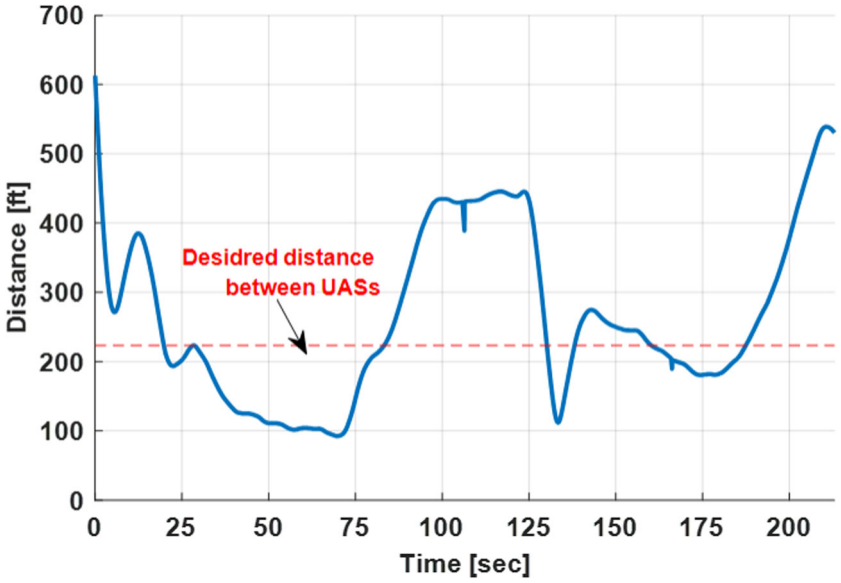

Fig. 12 Flight Test 1: Agent Separation during Autonomous Flight for DG808

initialization) and waypoint 2 (intersection to the racetrack by the average velocity of two agents during initialization) are temporarily assigned to stabilize the attitude angle at the beginning of the autonomous flight. Additionally, the altitude (top right) plot shows that one agent stayed at the desired higher altitude relative to the other agent according to safety measures taken during the flight test.

The first agent (blue) stayed inside of the race track while the second agent (red) stayed on the race track. Due to the confined area, the agents had to turn often leading to dramatic drops in the altitude and the rates of altitude ($\dot{h}$). Since the first agent followed the smaller race track pattern, it dropped altitude more significantly compared to the second agent due to larger aircraft bank angles.

Figure 9 presents the longitudinal states and controls for the first engagement of the autopilot. Figure 9, shows that although each UAS was engaged at **different initial conditions**, the GNC algorithms were robust toward out-of-trim aircraft and random initial conditions. The first agent is closer to the trim condition compared to the second agent since the pitch angle and rate have lower values. The second agent needs more time to settle down the attitudes after the autopilot is engaged. Additionally, the airspeed, pitch angle and rate of the first agent show a dramatic drop during the

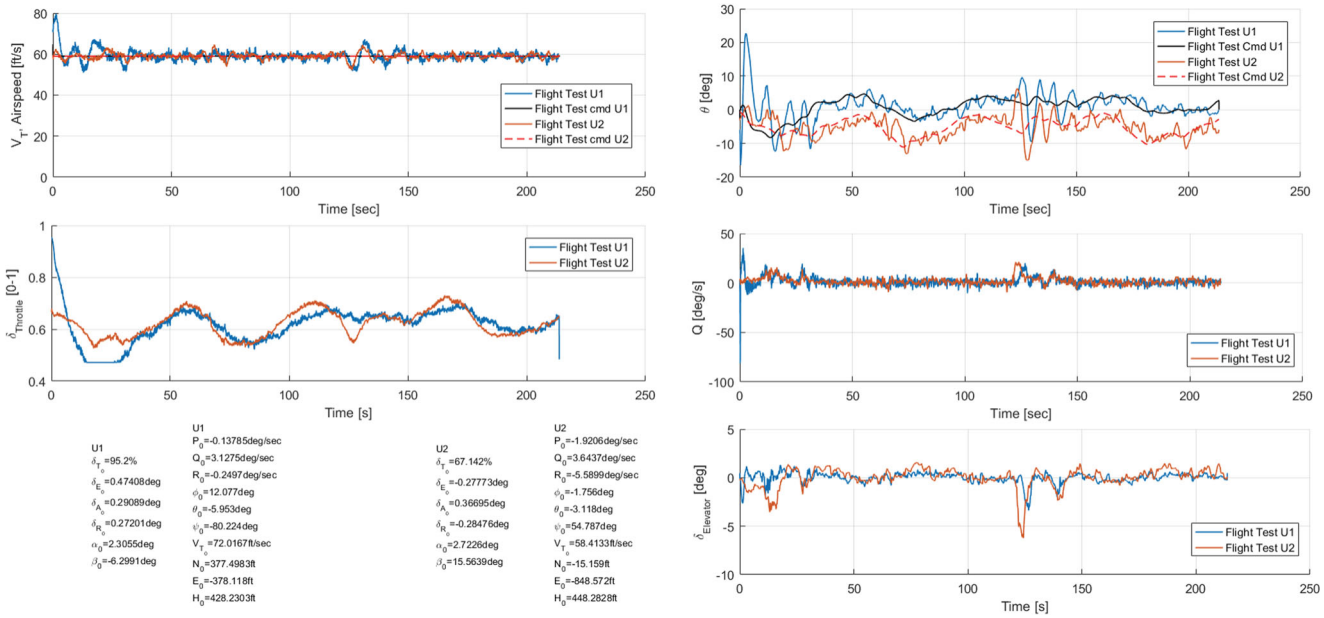

Fig. 13 Flight Test 1: 2nd autopilot engagement DG808 Longitudinal States

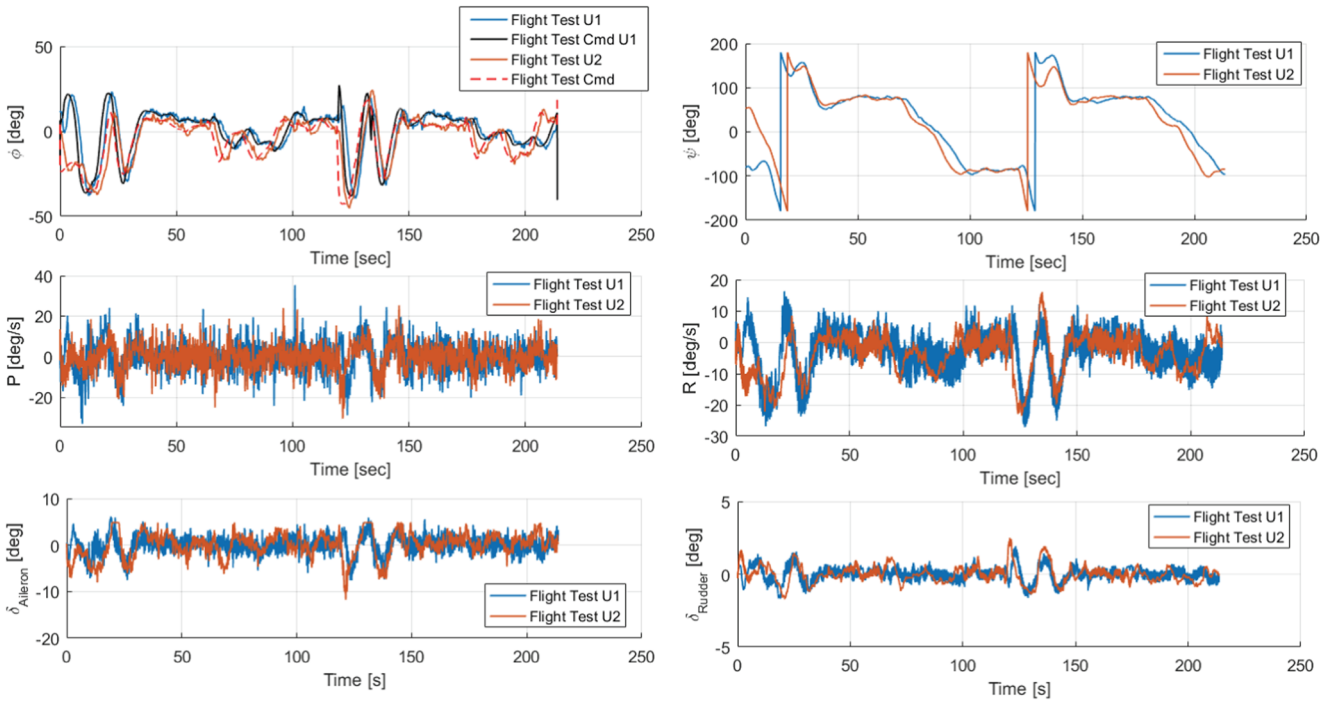

Fig. 14 Flight Test 1: 2nd autopilot engagement DG808 Lateral-Directional States

turning maneuvers since it had to follow the inner box of the race track (this induced a larger roll angle command) and this phenomenon is reflected in the altitude rate plots.

Figure 10 shows the lateral-direction states and controls for 1st engagement of the autopilot. The initial yaw angles of both agents were similar: 46 degree for the 1st agent

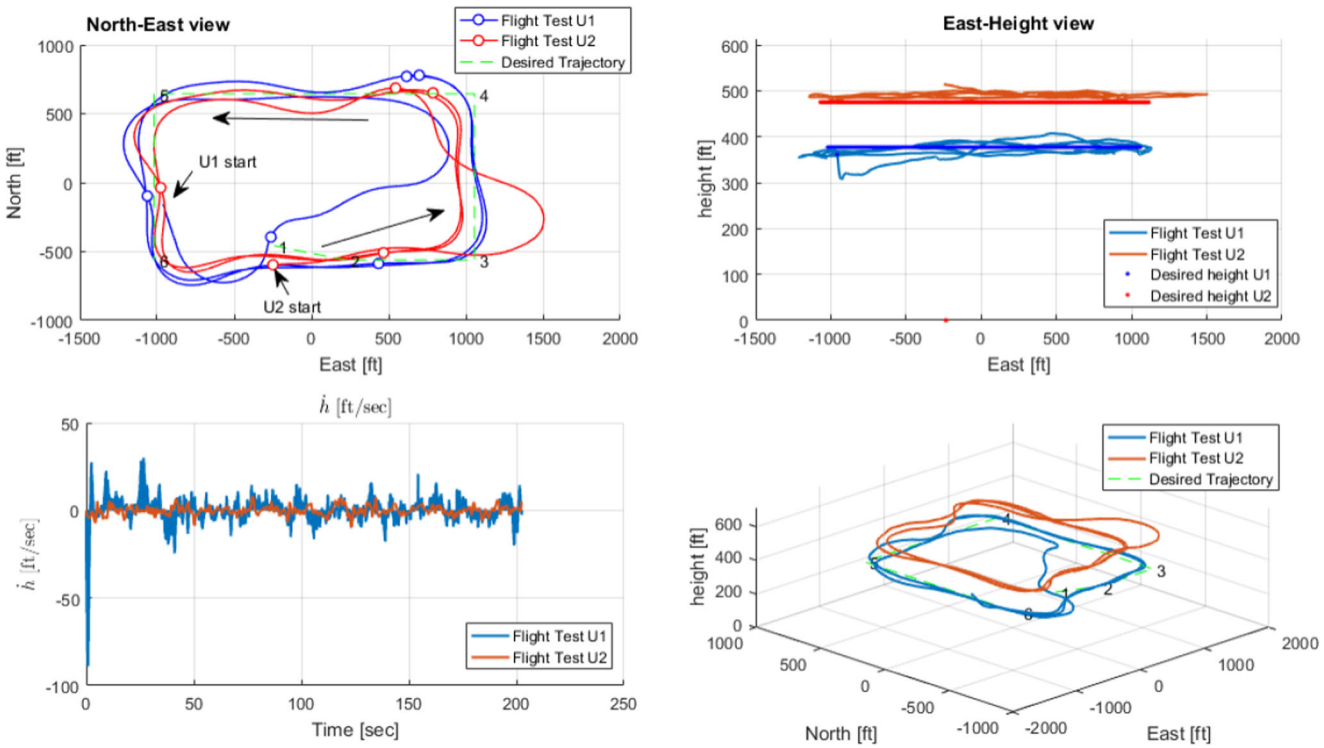

Fig. 15 Flight Test 2: Skyhunter Swarm Flight

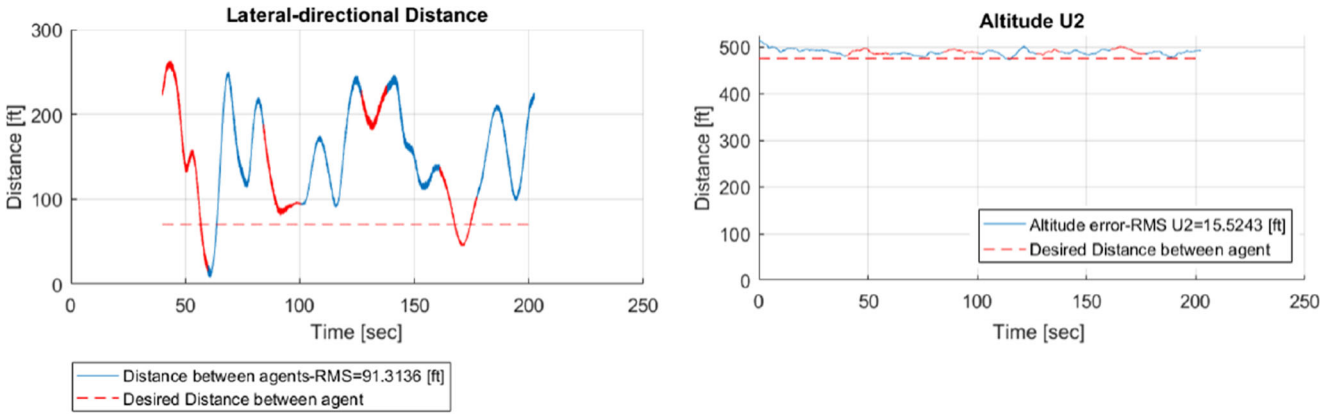

Fig. 16 Flight Test 2: Agent Separation during Autonomous Flight for Skyhunter

and 57 deg for the 2nd agent. However the second agent is closer to trim as compared to the first agent–if only lateral states are considered. The roll and yaw rate of the first agent is larger at the initial condition than for the second agent. The roll angles of the first agent is larger due to the smaller desired race track. The guidance and control algorithm worked in an appropriate way so that the agents could track the desired trajectories.

The second engagement of the autonomous flight has been presented below. The initial conditions are different and this test demonstrates that the developed algorithms have robustness towards randomness of initial conditions. In this test, the initial yaw angles are challenging: -80 degree for the first agent and 55 degree for the second agent. Basically, two agents were approaching each other. The position of two agents causes another challenge due to high moment of inertia and high speed. As Fig. 11 shows, the positions where autopilot were engaged at is close to the corner of the race track. This indicates that the agents should settle attitude and begin to turn very soon. Again, the morphing potential algorithm has been implemented and shows that the two agents successfully avoided collision with each other, see Ref. [14] as well. After 70 seconds of flight, the distance between the two agents was minimum during this flight and did not violate the minimum separation distance which is 100 ft, as shown

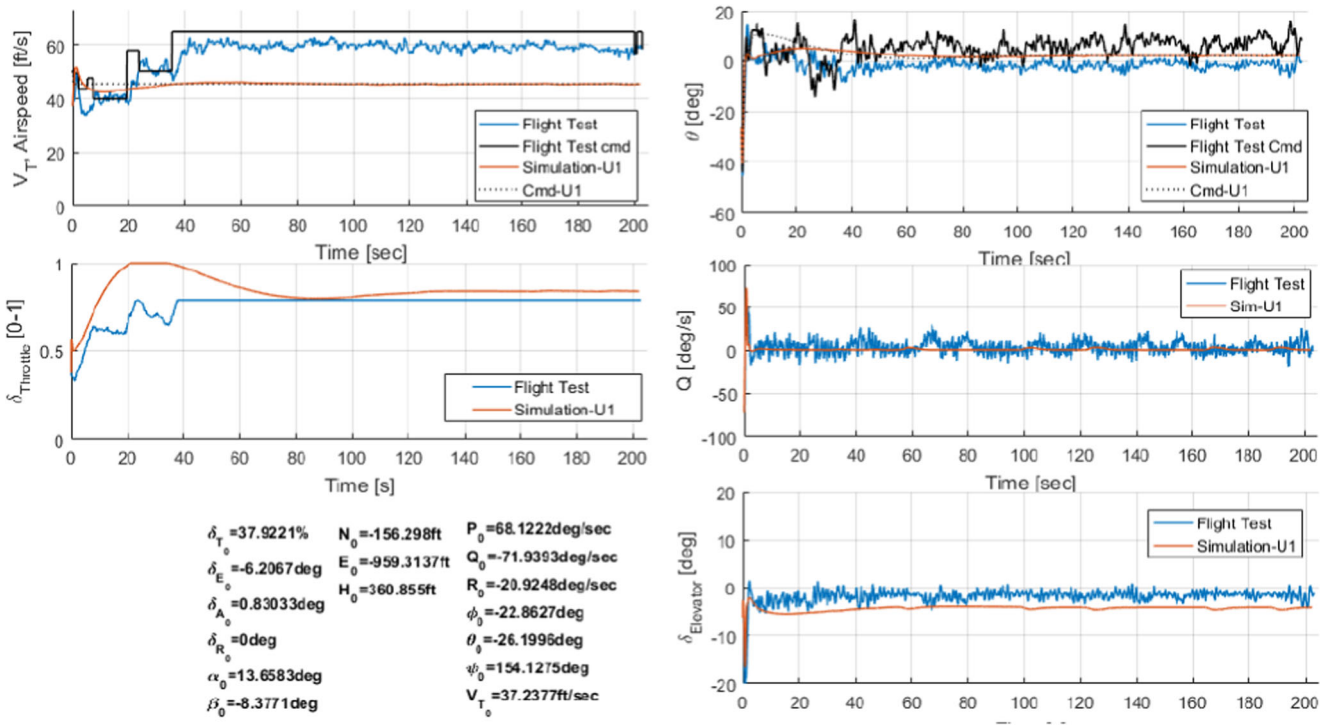

Fig. 17 Flight Test 2: Skyhunter Swarm U1 Longitudinal States

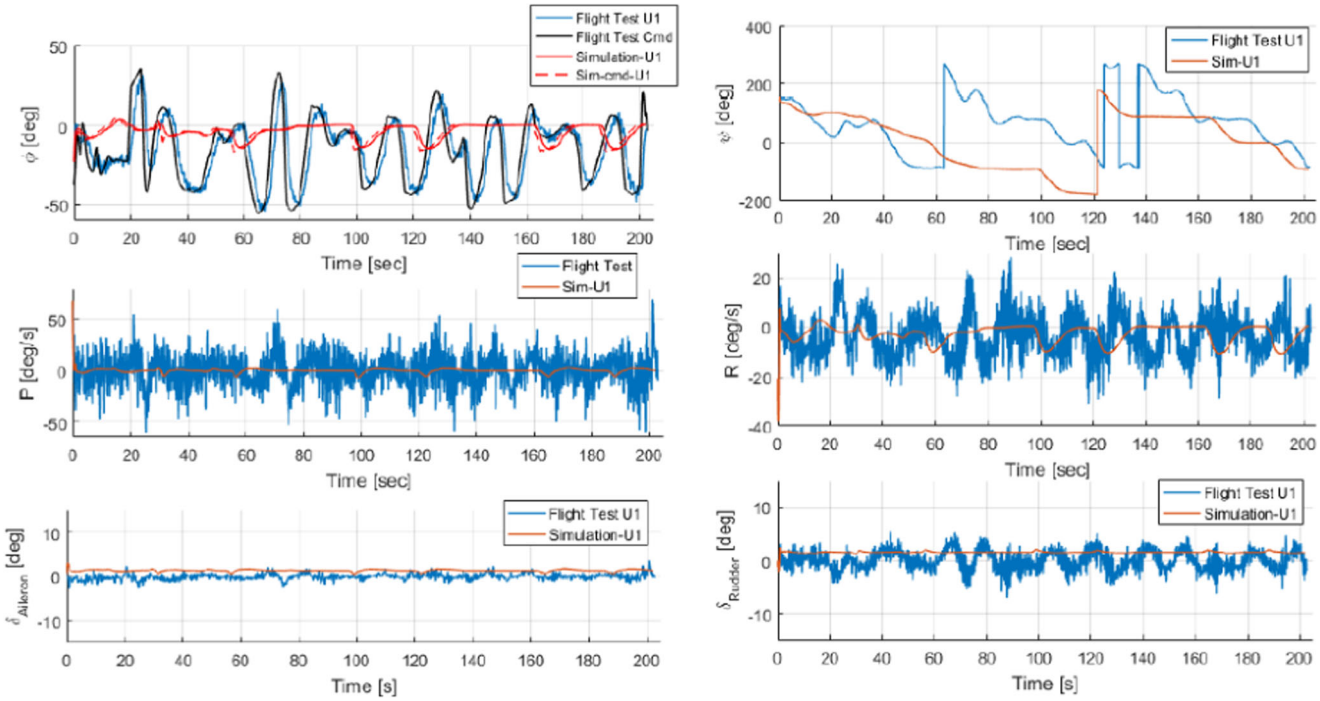

Fig. 18 Flight Test 2: Skyhunter Swarm U1 Lateral-Directional States

in Fig. 12. Additionally, Fig. 11 (top right) shows that the altitude of two agents were separated by the desired safe distance.

Figure 13 presents the longitudinal states and controls for the second engagement of the autonomous flight. The first agent has a large variation of the pitch angle, throttle, and airspeed. The initial airspeed was far from the trim point (it was 72 ft/sec), compared to 59.07 ft/sec. This led to dramatic changes in its airspeed, pitch angle and rate, and the altitude rate at the beginning of the autonomous flight.

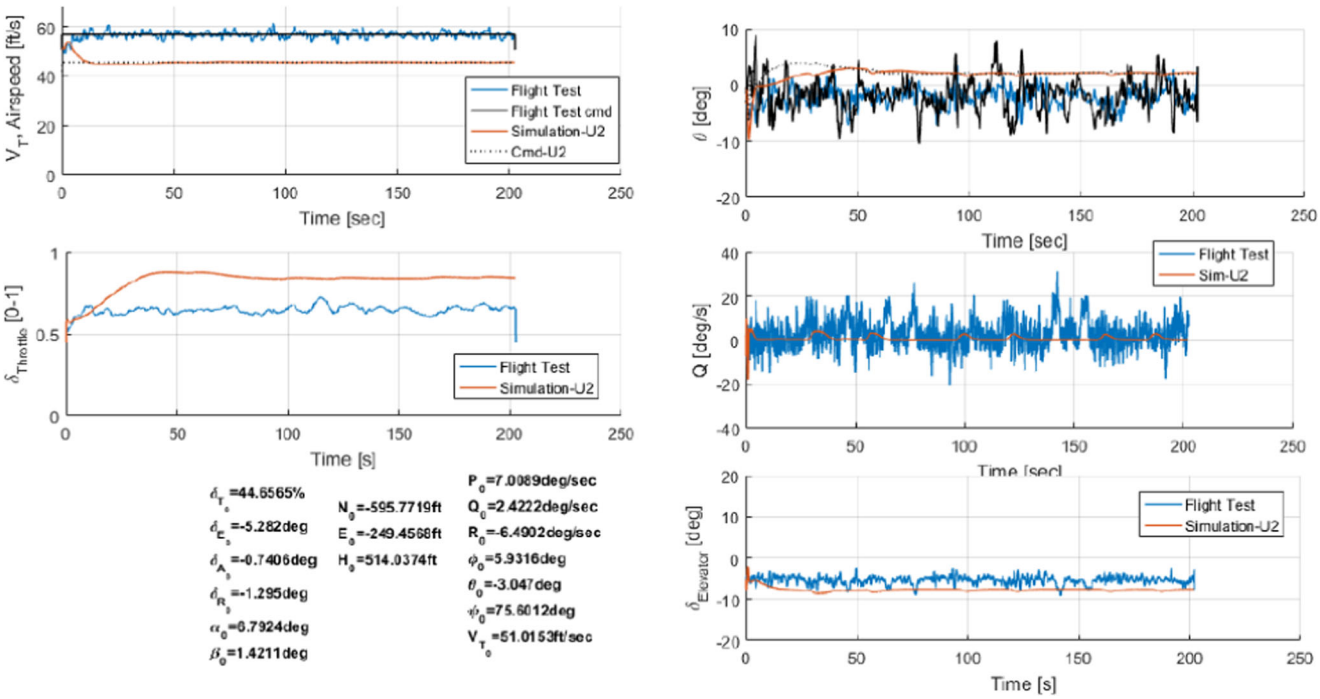

Fig. 19 Flight Test 2: Skyhunter Swarm U2 Longitudinal States

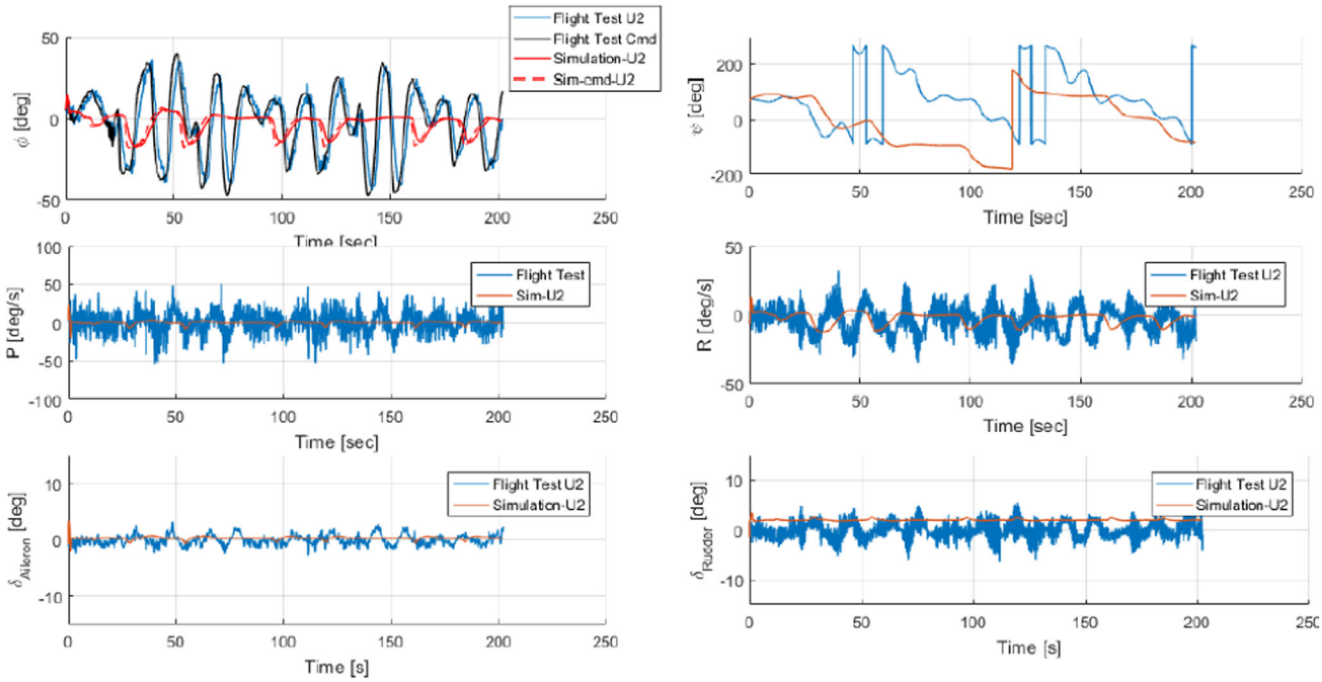

Fig. 20 Flight Test 2: Skyhunter Swarm U2 Lateral-Directional States

Next, the changes in pitch angle and rate and airspeed during the flight can be observed in the turning maneuver similar to the first engagement.

Figure 14 shows the lateral-directional states and controls for the second engagement of the autonomous flight. In this flight, the first agent was closer to the trim point since it has smaller roll rate and yaw rate compared to the second agent, although the initial roll angle of the first agent (12 deg) is larger than the second agent's (-1.7 deg). The roll angles of the second agent during turning maneuvers were larger than

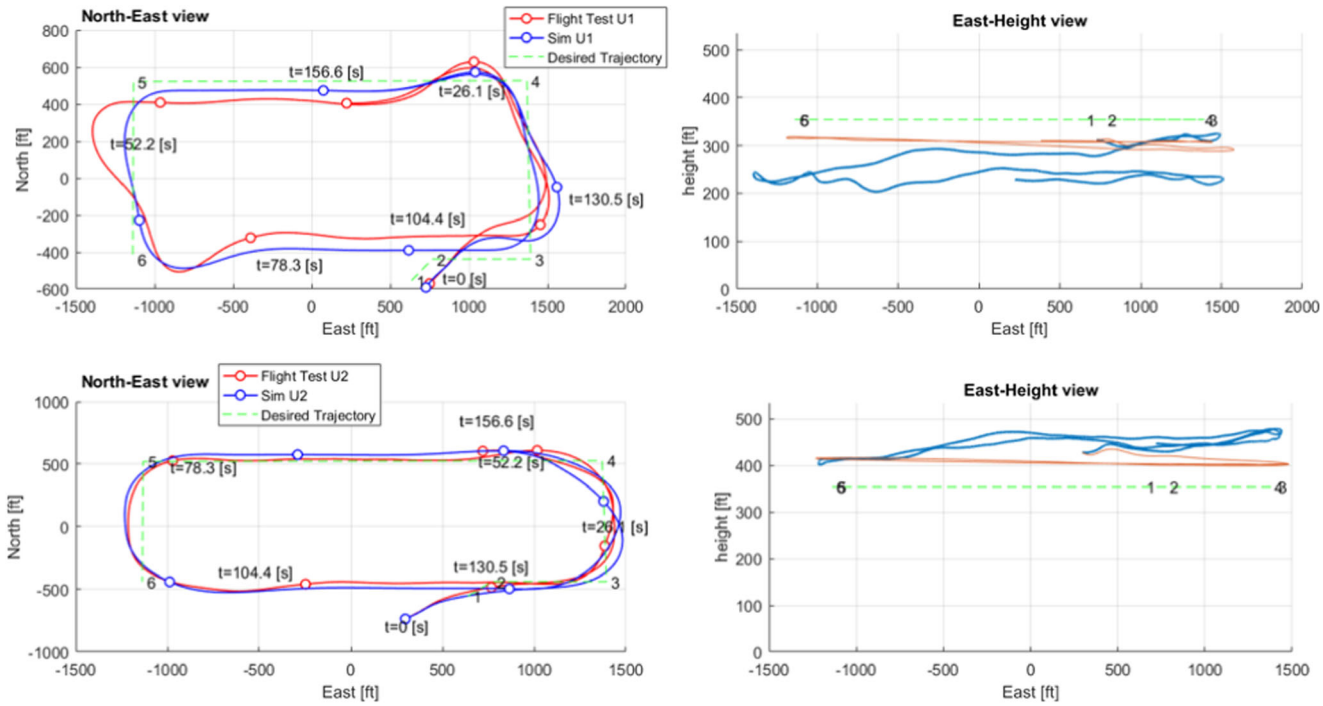

Fig. 21 Position comparison between the flight test (DG808) and simulation (Red: flight test, Blue: simulation)

the first agent due to the second agent following the inner race track pattern in this case.

Figure 15 shows the position of two Skyhunters during the flight test on April 26th, 2018. The desired formation distance was 50 ft separation to the North and 50 ft separation to the East between the agents. Each agent should stay at a unique altitude: 475 ft or 377 ft AGL, which is assigned based on the relative altitudes during autopilot initialization. Two agents initiated the swarm flight with different yaw angles (ψ): 154 and 76 degrees for agents 1 and 2, respectively. Next, the two agents followed the race track in the counterclockwise direction. The height tracking was tighter than that of the DG808 platforms, which were shown in Fig. 8.

As Figs. 12 and 15 show, when the SkyHunter UASs got closer than a safe distance (defined by collision avoidance parameters), **the morphing potential collision avoidance** kicked in and UAS2 avoided UAS1 (red trajectory on the South East corner), also see Ref. [14, 34]. Figure 16 shows the lateral and longitudinal direction tracking error. Compared to the DG808 (which has the total RMS of 129 ft), the total RMS has been improved by 29.2%. Even though the wind speed was higher comparing to the DG808 (average of 4.6 mph for DG808s and 10.4 mph for the Skyhunters during their respective flight tests), the tracking error is lower than the DG808. Figure 17 shows the longitudinal states and control of the Skyhunter agent 1.

The orange lines in the plots correspond to data from simulations performed using similar initial conditions to the flight test. The biggest difference between the simulation and the flight test data are the oscillations. The simulation environment does not incorporate the wind field, along with vibration due to the motor. Figure 18 shows the lateral-directional states and control of the Skyhunter agent 1.

Similarly, the agent drifted due to the wind and the roll angle commands are larger than the simulation. Additionally, oscillations in roll and yaw rates are observed. Due to the oscillatory behavior, the control surface deflections also reacted to these rates.

Figure 19 shows the longitudinal states and control of the Skyhunter agent 2. Similarly, oscillations in the pitch rate led to oscillations in the commanded elevator deflection. This relation is also seen in the pitch angle when compared to that of the simulation. Figure 19 shows the lateral-directional states and controls of Skyhunter agent 2. This is also similar to Fig. 18. The roll angles are larger than the simulation due to the agent being drifted by the wind.

The results of these flight tests involved the two Skyhunters holding the desired formation using identical GNC algorithms to the ones used for the DG808, using only different gains for the guidance and control parameters. Based on the RMS of the tracking error, the Skyhunter achieved better performance than the DG808 even though the wind speed was higher for the Skyhunter flight tests.

7 Comparison of Flight test and Simulations

This section presents the comparison between the flight test and simulation data.

Figures 20, 21, 22, 23, 24 and 25 show comparison of flight test demonstrations versus simulation results. Waypoints, desired flight paths, and initial conditions were identical. Although simulations are imperfect representative

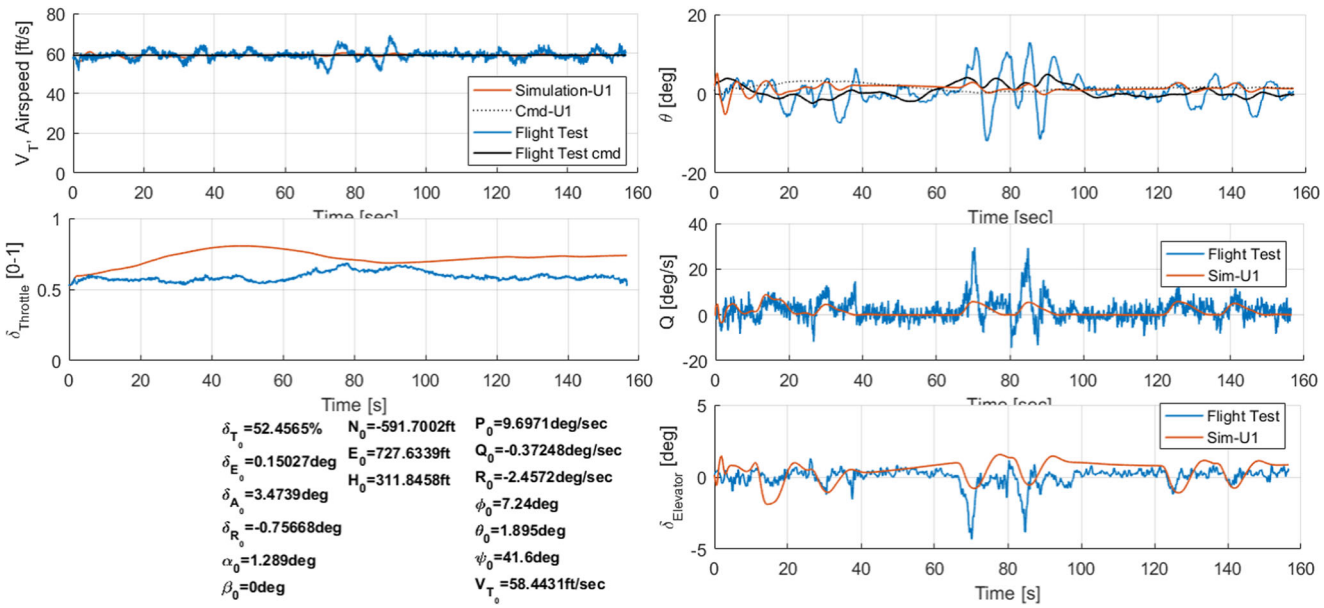

Fig. 22 Longitudinal states (U1) comparison between the flight test (DG808) and simulation (Blue: flight test, Red: simulation)

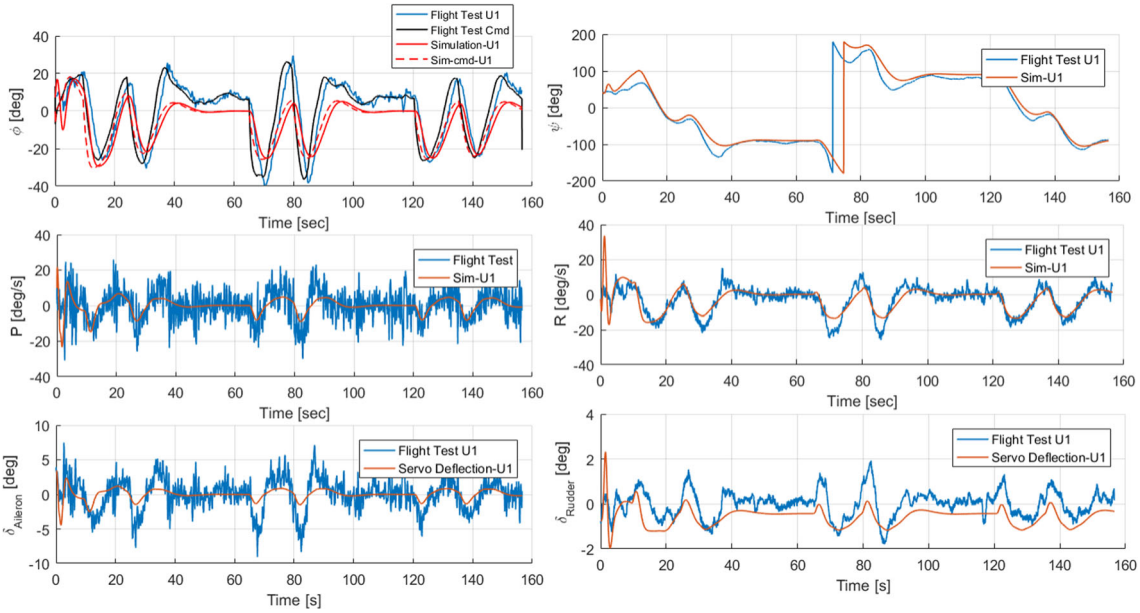

Fig. 23 Lateral states (U1) comparison between the flight test (DG808) and simulation (Blue: flight test, Red: simulation)

of reality, a good agreement can be observed between simulation and flight test results. However, uncertainties in the dynamic model of UASs due to the low-fidelity of engineering level dynamic analysis software, lack of robustness in the LQR controller, sensor noises and external disturbances directly influence credibility of simulations and cause mismatches between simulation and flight test results. Five important lessons learned are summarized as follows: (1) UASs' altitude and velocity hold controllers worked very effectively and flight test and simulation results match very well (see Figs. 21, 22, and 24). (2) UASs path following performance in flight tests was similar to simulation results however the flight test was conducted in the presence 4.6 mph West wind which affected the trajectory of UASs on corners. (3) Impact of sensor noises and motor vibration are evident on rate gyros and velocity measurements. (4) Impact of external disturbances (e.g. wind, etc.) can be observed in uncommanded motions and delay in UASs' positions and consequently guidance commands. (5) Comparison of results show the importance

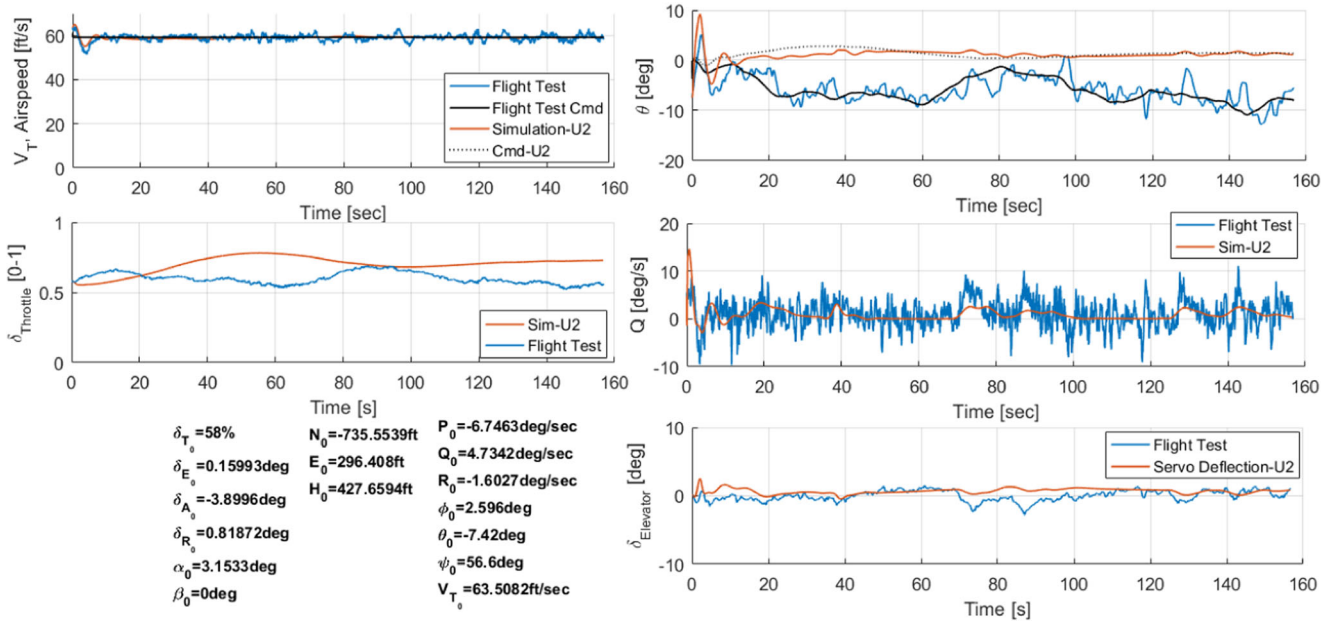

Fig. 24 Longitudinal states (U2) comparison between the flight test (DG808) and simulation (Blue: flight test, Red: simulation)

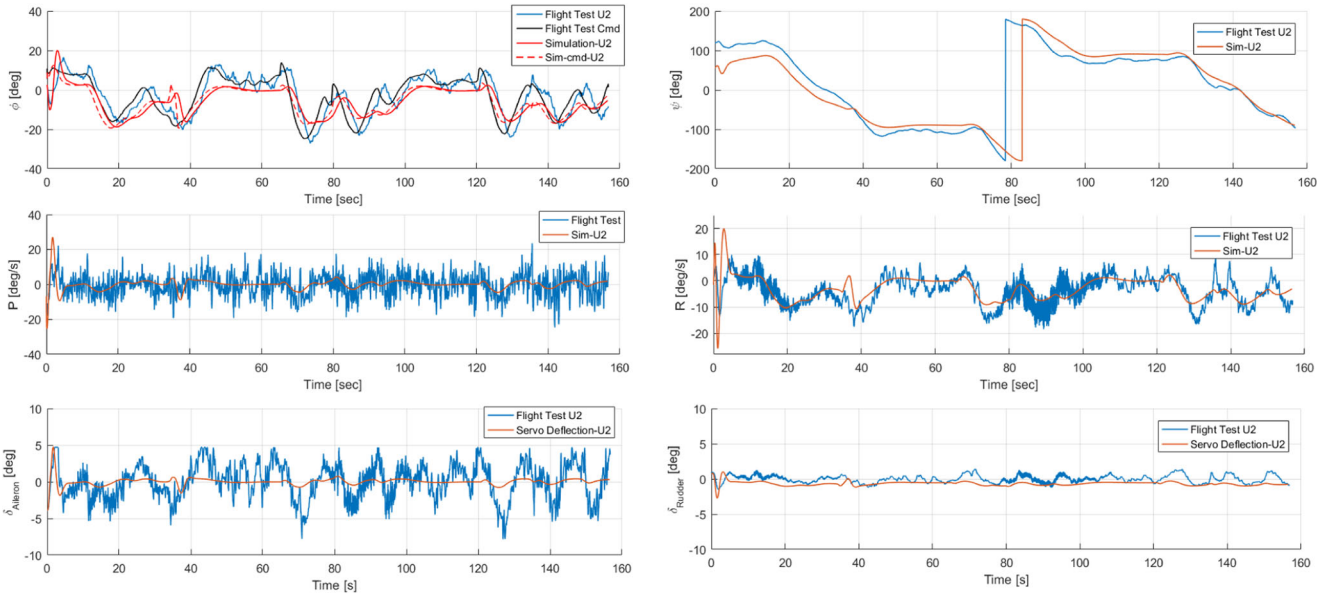

Fig. 25 Lateral states (U2) comparison between the flight test (DG808) and simulation (Blue: flight test, Red: simulation)

of robustness of flight controllers (e.g. H-inf or H2 controllers) to mitigate effect of external disturbances, uncertainty in dynamic models, and noise and vibration when UASs perform complex missions in unstructured environments.

8 Conclusion

A complete set of guidance, navigation and control algorithms were developed for collaborative UASs and uniquely flight tested using two completely different UAS platforms. The developed GNC algorithms demonstrated their effectiveness in holding a desired separation distance between large collaborative UAS flying at high speeds. Robustness of GNC algorithms was successfully validated using out-of-trim or random initial conditions. Although, the desired distance between UASs was reduced in the second round of validation and verification flight test, the UASs held their positions in the formation. Additionally, when the distance between agents crossed the defined threshold, the collision avoidance algorithms updated the aircraft trajectory in order to achieve the desired separation. Comparison of simulation and flight test results are presented and the impact of modeling uncertainty, noise and disturbances are identified.

Acknowledgements This work was completed with funding from NASA Learn project #NNX15AN94A and NASA CAN project #NNX15AN04A at the University of Kansas. The authors greatly appreciate this NASA support. In addition, the authors would like to thank our UAS pilot Matt Tener and KUAE undergraduate student Grant Godfrey for all of their support during flight testing operations. The author appreciates the help from Dr. Gonzalo Garcia for building the simulation environment.

References

1. Alonso-Mora, J., Breitenmoser, A., Rufli, M., Siegwart, R., Beardsley, P.: Multi-robot system for artistic pattern formation. In: 2011 IEEE international conference on robotics and automation, pp. 4512–4517. IEEE (2011)
2. Alonso-Mora, J., Breitenmoser, A., Rufli, M., Siegwart, R., Beardsley, P.: Image and animation display with multiple mobile robots. Int. J. Robot. Res. **31**(6), 753–773 (2012)
3. Bayezit, I., Fidan, B.: Distributed cohesive motion control of flight vehicle formations. IEEE Trans. Ind. Electron. **60**(12), 5763–5772 (2013)
4. Bayraktar, S., Fainekos, G.E., Pappas, G.J.: Experimental cooperative control of fixed-wing unmanned aerial vehicles. In: 2004 43rd IEEE conference on decision and control (CDC)(IEEE Cat. No. 04CH37601), vol. 4, pp. 4292–4298. IEEE (2004)
5. Boskovic, J.D., Li, S.M., Mehra, R.K.: Formation flight control design in the presence of unknown leader commands. In: Proceedings of the 2002 American control conference (IEEE Cat. No. CH37301), vol. 4, pp. 2854–2859. IEEE (2002)
6. Chuang, Y.L., Huang, Y.R., D'Orsogna, M.R., Bertozzi, A.L.: Multi-vehicle flocking: scalability of cooperative control algorithms using pairwise potentials. In: Proceedings 2007 IEEE international conference on robotics and automation, pp. 2292–2299. IEEE (2007)
7. Do, K.D.: Coordination control of multiple ellipsoidal agents with collision avoidance and limited sensing ranges. Syst. Control Lett. **61**(1), 247–257 (2012)
8. Garcia, G., Keshmiri, S.: Nonlinear model predictive controller for navigation, guidance and control of a fixed-wing uav. In: AIAA guidance, navigation, and control conference, guidance, navigation, and control and Co-located conferences (2011)
9. Gilles, J., Sharma, B.R., Ferenc, W., Kastein, H., Lieu, L., Wilson, R., Huang, Y.R., Bertozzi, A.L., HomChaudhuri, B.,

Ramakrishnan, S., et al.: Robot swarming over the internet. In: 2012 American control conference (ACC), pp. 6065–6070. IEEE (2012)
10. Gonzalez, M., Huang, X., Martinez, D.S.H., Hsieh, C.H., Huang, Y.R., Irvine, B., Short, M.B., Bertozzi, A.L.: A third generation micro-vehicle testbed for cooperative control and sensing strategies. In: ICINCO (2), pp. 14–20 (2011)
11. Huang, W., Russell, R.D.: Adaptive moving mesh methods. Springer, New York (2011)
12. Hui, M.Y., Yang, C.Q., Xi, H.Z., Zheng, G.: An improved nonlinear guidance law for unmanned aerial vehicles path following. In: 2015 34th Chinese control conference (CCC), pp. 5271–5276 (2015). https://doi.org/10.1109/ChiCC.2015.7260462
13. Keshmiri, S., Kim, A., Blevins, A., Shukla, D.: Validation and verification flight test of uas morphing potential field collision avoidance algorithms. AIAA aviation and aeronautics forum and exposition (2018)
14. Keshmiri, S., Kim, A.R., Shukla, D., Blevins, A., Ewing, M.: Flight test validation of collision and obstacle avoidance in fixed-wing uass with high speeds using morphing potential field. In: 2018 international conference on unmanned aircraft systems (ICUAS), pp. 589–598 (2018). https://doi.org/10.1109/ICUAS.2018.8453299
15. Kim, A.R., Keshmiri, S., Huang, W., Garcia, G.: Guidance of multi-agent fixed-wing aircraft using a moving mesh method. Unmanned Systems **04**(03), 227–244 (2016). https://doi.org/10.1142/S2301385016500084
16. Lin, Z., Ding, W., Yan, G., Yu, C., Giua, A.: Leader–follower formation via complex laplacian. Automatica **49**(6), 1900–1906 (2013)
17. Liu, W., Taima, Y.E., Short, M.B., Bertozzi, A.L.: Multi-scale collaborative searching through swarming. In: ICINCO (2), pp. 222–231 (2010)
18. Mesbahi, M.: On state-dependent dynamic graphs and their controllability properties. IEEE Trans. Autom. Control **50**(3), 387–392 (2005)
19. Michael, N., Mellinger, D., Lindsey, Q., Kumar, V.: The grasp multiple micro-uav testbed. IEEE Robot. Autom. Mag. **17**(3), 56–65 (2010)
20. Nguyen, B.Q., Chuang, Y.L., Tung, D., Hsieh, C., Jin, Z., Shi, L., Marthaler, D., Bertozzi, A., Murray, R.M.: Virtual attractive-repulsive potentials for cooperative control of second order dynamic vehicles on the caltech mvwt. In: Proceedings of the 2005, American control conference, 2005., pp. 1084–1089. IEEE (2005)
21. Nigam, N., Bieniawski, S., Kroo, I., Vian, J.: Control of multiple uavs for persistent surveillance: algorithm and flight test results. IEEE Trans. Control Syst. Technol. **20**(5), 1236–1251 (2012)
22. Olfati-Saber, R., Murray, R.M.: Graph rigidity and distributed formation stabilization of multi-vehicle systems (2002)
23. Park, S., Deyst, J., How, J.: A new nonlinear guidance logic for trajectory tracking. In: AIAA guidance, navigation, and control conference and exhibit, guidance, navigation, and control and Co-located conferences, pp. 1–16 (2004)
24. Qiu, J., Sun, K., Wang, T., Gao, H.: Observer-based fuzzy adaptive event-triggered control for pure-feedback nonlinear systems with prescribed performance. IEEE Trans. Fuzzy Syst. 1–1 (2019). https://doi.org/10.1109/TFUZZ.2019.2895560
25. Ratnoo, A., Sujit, P., Kothari, M.: Adaptive optimal path following for high wind flights. In: Preprints of the 18th IFAC world congress, pp. 12985–12990 (2011)
26. Ren, W., Beard, R.W.: Consensus seeking in multiagent systems under dynamically changing interaction topologies. IEEE Trans. Autom. Control **50**(5), 655–661 (2005)
27. Ren, W., Beard, R.W.: Consensus seeking in multiagent systems under dynamically changing interaction topologies. IEEE Trans. Autom. Control **50**(5), 655–661 (2005)
28. Ren, W., Cao, Y.: Distributed coordination of multi-agent networks: emergent problems, models, and issues. Springer Science & Business Media (2010)
29. Rezaee, H., Abdollahi, F.: A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots. IEEE Trans. Ind. Electron. **61**(1), 347–354 (2014)
30. Roskam, J.: Airplane flight dynamics and automatic flight controls Part I. DARcorporation (2003)
31. Saber, R.O., Dunbar, W.B., Murray, R.M.: Cooperative control of multi-vehicle systems using cost graphs and optimization. In: Proceedings of the 2003 American control conference, 2003., vol. 3, pp. 2217–2222. IEEE (2003)
32. Semsar-Kazerooni, E., Khorasani, K.: Optimal consensus algorithms for cooperative team of agents subject to partial information. Automatica **44**(11), 2766–2777 (2008)
33. Semsar-Kazerooni, E., Khorasani, K.: An optimal cooperation in a team of agents subject to partial information. Int. J. Control. **82**(3), 571–583 (2009)
34. Stastny, T.J., Garcia, G., Keshmiri, S.: Robust three-dimensional collision avoidance for fixed-wing unmanned aerial systems. https://arc.aiaa.org/doi/abs/10.2514/6.2015-1988
35. Stevens, B.L., Lewis, F.L.: Aircraft control and simulation. J Wiley (2003)
36. Sun, K., Mou, S., Qiu, J., Wang, T., Gao, H.: Adaptive fuzzy control for non-triangular structural stochastic switched nonlinear systems with full state constraints. IEEE Trans. Fuzzy Syst. 1–1 (2018). https://doi.org/10.1109/TFUZZ.2018.2883374
37. United States Government Accountability Office: Defense acquisitions assessments of selected weapon programs. https://www.gao.gov/assets/660/653379.pdf (2013)
38. Xi, X., Abed, E.H.: Formation control with virtual leaders and reduced communications. In: Proceedings of the 44th IEEE conference on decision and control, pp. 1854–1860. IEEE (2005)
39. Xiao, J., Zhang, J., Adler, B., Zhang, H., Zhang, J.: Three-dimensional point cloud plane segmentation in both structured and unstructured environments, vol. 61. http://www.sciencedirect.com/science/article/pii/S0921889013001152 (2013)
40. Xiao, J., Zhang, J., Zhang, J., Zhang, H., Hildre, H.P.: Fast plane detection for slam from noisy range images in both structured and unstructured environments. In: 2011 IEEE international conference on mechatronics and automation, pp. 1768–1773 (2011). https://doi.org/10.1109/ICMA.2011.5986247
41. Zheping, Y., Yibo, L., Jiajia, Z., Gengshi, Z.: Moving target following control of multi-auvs formation based on rigid virtual leader-follower under ocean current. In: 2015 34th Chinese control conference (CCC), pp. 5901–5906. IEEE (2015)
42. Zunli, N., Xuejun, Z., Xiangmin, G.: Uav formation flight based on artificial potential force in 3d environment. In: 2017 29th Chinese control and decision conference (CCDC), pp. 5465–5470. IEEE (2017)