Flight Test Validation of Collision and Obstacle Avoidance in Fixed-Wing UASs with High Speeds Using Morphing Potential Field

A Ram (Bella) Kim, Iowa State University
Flight Test Validation of Collision and Obstacle Avoidance in Fixed-Wing UASs with High Speeds Using Morphing Potential Field

Shawn Keshmiri  A Ram Kim  Daksh Shukla  Aaron Blevins  Mark Ewing
University of Kansas
Department of Aerospace Engineering
Lawrence, 66045, KS

Abstract—A novel approach to collision and obstacle avoidance in fixed-wing unmanned aerial systems with high speed and high inertia was developed by reformulating classical artificial potential field navigational approaches. Classical artificial potential field navigation is a formidable approach to collision avoidance for slow and small robots including rotary-wing UASs, however they lack robustness and adaptability for large fixed-wing aircraft flying in close proximity or congested areas. As part of a concept demonstration, this work presents the validation and verification of morphing potential collision avoidance using large unmanned aerial systems flying at 60 ft/sec. The morphing potential function was constrained by the aircraft’s six-degree-of-freedom dynamic characteristics and maximum allowable bank angle. A virtual time-varying waypoint is used to navigate the aircraft in a dynamically changing environment. The validation flight tests were successfully conducted and real-time avoidance capabilities were demonstrated.

Index Terms—Morphing Potential Field, Collision Avoidance

I. INTRODUCTION

The next generation of UASs must be capable of collision-free operations in congested urban environments. UASs must be capable of “see and avoid” for an arbitrary number of fixed obstacles and moving objects, including collaborative and non-collaborative aircraft. This means the “avoidance” tasks could include much more proximal operations where vehicles must be able to maneuver quickly and efficiently, on short notice, in the event of encountering non-collaborative aircraft or onboard adverse conditions (e.g. system failures) on neighboring aircraft.

The Federal Aviation Administration (FAA) demands unmanned aerial systems (UASs) to be compliant with the “see and avoid” provisions of Code 14 of Federal Regulations (CFR) 91.113, Right-of-way rules. For transportation aircraft, a five nautical mile separation distance is required by the FAA, however currently there is no concrete distance requirement for UASs. In the near future, the U.S. will adapt a hybrid national airspace (manned and unmanned). This fact has sparked a large amount of interest into the problem of UAS air traffic control (ATC) and collision and obstacle avoidance in autonomous aircraft operations [1]–[4]. The artificial potential field (APF) methods have been widely implemented on a variety of applications with much success for collision and obstacle free path planning and control in robotics [4]–[6] and over the past few years have been applied in multi-agent UAS guidance [1]–[3], [7], [8], however the majority of existing works are applied for relatively slow moving robots/cars [4]–[6], [9], or rotary-wing UASs [10]–[12] with the exception of a few works on fixed-wing UASs [13], [14]. Another common denominator of existing works is simplified aerody-
namic models and relaxed dynamical constraints. A large amount of research assume aircraft as point-masses models where aircraft can perform instantaneous 90 degrees turns \([2], [3], [15]–[17]\). The point mass assumption is acceptable as long as agents are moving slowly, however such assumptions cannot be applied to large and fixed-wing USAs flying at high speeds.

Previously, our team developed a novel approach to collision and obstacle avoidance in fixed-wing unmanned aerial systems using adaptively morphing APF. The new method was successfully used for vehicles with high speed and high inertia, operating in proximal or congested settings in simulation environment \([18]\). This work presents the actual implementation of this navigation algorithm and actual validation and flight test of USAs.

**II. SUMMARY OF MORPHING POTENTIAL FIELD ALGORITHMS**

This section only shows the high-level mathematical algorithms, and more details can be found in reference \([18]\). Gaussian-shaped potential field formulations use the norm of distance between the agent and another agent or obstacle, where the origin of the field is placed at the centroid of the agent/object to be avoided. Equation 1 shows the standard potential field where the subscript "obj" represents the agent(s) or obstacle(s) to avoid.

\[
pf = A \cdot \exp \left\{ - \frac{\left( \| \mathbf{p}^{\text{obj}} - \mathbf{p}^{o} \| \right)^{2}}{\sigma} \right\}
\]  

\(\sigma\) represents the desired avoidance radius and \(A\) is a proportional gain to increase or decrease the amplitude, and \(\mathbf{p}^{\text{obj}}\) and \(\mathbf{p}^{o}\) are defined as the 2-D lateral inertial position coordinates (North and East) of the object to avoid. Considering the structural limits of fixed-wing aircraft and its minimum turning radius, the avoidance maneuvers must be planned and executed further in advance. Considering the circular shape of classical APF, the resulting evasion path is fairly inefficient as the aircraft after passing and successfully avoiding the obstacle, will maintain the same and unnecessary distance with the obstacle due to the potential field’s symmetry. To remedy these issues, Ref. \([18]\) proposed a “morphing” factor \(\Gamma\), which was integrated into the potential function (see Eq. 2). Morphing potential is a function of approach angle, aircraft velocity, and relative velocity \(\vec{v}^{o} - \vec{v}^{\text{obj}}\). An additional reference shifting term \(\vec{S}\) has also been included in the distance norm as a means of further shaping the potential to avoid unnecessary levels of cost beyond the avoided obstacle by shifting the potential function origin \(\vec{S}\) away from the centroid of the object. The resultant formulation is deemed a morphing potential function \(mpf\).

\[
mpf = \exp \left\{ -\Gamma \left( \frac{\| \mathbf{p}^{\text{obj}} - \mathbf{p}^{o} + \vec{S} \|}{\sigma} \right)^{2} \right\}
\]  

The reference shifting term \(\vec{S}\) is defined from the obstacle position \(\mathbf{p}^{\text{obj}}\) in the direction opposite the relative moving point velocity \(\vec{v}^{o}\) some distance \(d_s = f(R_{\text{min}})\), i.e. \(\vec{S} = \frac{\vec{v}^{o}}{\|\vec{v}^{o}\|}d_s\), where \(d_s \leq R_{\text{min}}\). \(R_{\text{min}}\) is the minimum turning radius of the fixed-wing USAS aircraft which is a function of maximum allowed bank angle \(\phi_{\text{max}}\) and cruise speed (see Eq. 3)

\[
R_{\text{min}} = \frac{\| \vec{v}^{o} \|^{2}}{g \tan \phi_{\text{max}}}
\]  

The morphing factor is expressed in Eq. 4:

\[
\Gamma = (1 - \Gamma) \sin^{2} \eta_v + \Gamma_0; \quad \Gamma_0 \in (0, 1], \quad \eta_v \in [-\pi, \pi]
\]  

where \(\eta_v\) is the angle between the moving point relative velocity vector, \(\vec{v}^{o}\) and the distance vector and \(\Gamma_0\) is the extension term of the morphing factor. The potential factor \(\Gamma\) can change the avoidance distance \(\sigma\). Depending on the relative velocity and approach direction morphing or shifting occurs and the function reduces to the general potential formulation.

\[
\Gamma_0 = \begin{cases} 
\left( \frac{\sigma}{\sigma + R_{\text{min}} - d_s} \right)^{2}, & |\eta_v| < \pi/2 \\
\left( \frac{\sigma}{\sigma + d_s} \right)^{2}, & \text{else}
\end{cases}
\]
With the potential field reformulated, calculating the gradient (see Eq. 6), negating, and normalizing (i.e. forming a unit vector, except where a gradient of zero gives a zero vector) will produce a vector field usable in the moving point planning algorithm.

\[
\nabla mpf = mpf \cdot \frac{2}{\sigma^2} \left\{ \frac{\vec{d}_c}{\| \vec{d}_c \|} - \frac{(1 - \Gamma_0) \cos \eta v}{\| \vec{d}_c \|} \frac{\vec{v}}{\| \vec{v} \|} \right\}
\]

(6)

Inspired by work done by Chang et. al. in [19] and to avoid the common deadlock problem in APF navigational approaches where UASs will randomly pick a direction to perform the avoidance maneuvers, the superposition of a gyroscopic vector field (see Eq. II) is used. The origin of the gyroscopic field is placed at \( \vec{c} \) and given a rotation direction (either clockwise or counter-clockwise). Finding the correct weighting necessary on the gyroscopic field is not a trivial job, and this value changes between UAS platforms. It is mainly a function of UAS maneuverability and acceptable avoidance scenarios.

\[
gyro = [-d_{cE}, d_{cN}]
\]

(7)

where \( d_{cN} \), \( d_{cE} \) are the North and East inertial components of the distance vector, \( \vec{d}_c \). Note the formulation shown in Eq. II produces a clockwise gyration, and that negating the vector field will give a counter-clockwise gyration. Following the avoidance maneuver, the aircraft returns to the desired heading (e.g. trajectory) according to an arctangent function (Eq. 8). \( \xi_{traj} \) represents the desired heading and the coordinate point \( \vec{p}_{traj} \) can be any point on the desired trajectory path. This function generates a vector field steering moving points asymptotically to the desired path. The normalized output of the function is shown as a vector field.

\[
\xi_{app} = \frac{2}{\pi} \xi_{app_{max}} \tan^{-1}\left\{ K_{traj} \left[ \left( \vec{p}_N^{\sigma} - \vec{p}_{traj}^{N} \right) \sin \xi_{traj} - \left( \vec{p}_E^{\sigma} - \vec{p}_{traj}^{E} \right) \cos \xi_{traj} \right] \right\}
\]

(8)

where \( \vec{p}_{traj}^{N} \) and \( \vec{p}_{traj}^{E} \) are the North and East position coordinates of any arbitrary point on the straight trajectory path, \( \xi_{app} \) is the approach heading relative to the trajectory path, \( \xi_{app_{max}} \) is the maximum approach heading (user defined), and \( K_{traj} \) is a gain adjusting the shape of approach vectors. Large values for \( \xi_{app_{max}} \) will result in a faster return to the desired trajectory, however this can lead to aggressive maneuvers which is not desirable or necessary in most cases.

III. DYNAMIC MODEL OF DG-808 UAS

The aircraft used in this research is the DG808S UAS, a sailplane model aircraft with a fiberglass fuselage and balsa wood wings. The aircraft is 1.56 meters in length, has a 4 meter wingspan, and has a mass of 4.4 kg including all avionics component. The cruise speed for the aircraft is 35 knots, and the stall speed is 18 knots. The aircraft is powered by an electric engine using an electronic speed controller (ESC), and has servos that actuate the elevator, rudder, ailerons, and flap control surfaces. The aircraft during a flight test is shown below in Figure 1. In this project, two DG808 aircraft with identical airframes and avionic components are used.

Fig. 1. DG-808 UAS
In order to determine the aircraft moments of inertia, a bifilar pendulum test was conducted on the full aircraft with all avionics on-board (see [20]). The resulting moment of inertia estimates are shown in Table I. The $I_{zz}$ component of the aircraft was assumed to be negligible due to the relative magnitude being of small numerical value.

<table>
<thead>
<tr>
<th>Moment of Inertia Kg-m$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{xx}$</td>
</tr>
<tr>
<td>$I_{yy}$</td>
</tr>
<tr>
<td>$I_{zz}$</td>
</tr>
</tbody>
</table>

To characterize the motor dynamics of the system, engine testing was conducted at the Mal Harned Propulsion Lab at the Garrison Flight Research Center in Lawrence, Kansas. This provided the static thrust values for the engine across its RPM range, as well as correlated the static thrust with the input pulse width modulation (PWM) for controller design. These static thrust values were then used to estimate the in-flight thrust values at the cruising velocity of 35 knots using methods proposed in References [21] that convert thrust values for a similar propeller across varying advance ratios [22]. Using the moment of inertia estimates and the motor dynamics, two different aircraft modeling approaches were used to obtain and verify the values of aircraft stability and control derivatives. The first of which was Advanced Aircraft Analysis (AAA), a product of DARcorporation, which uses aircraft geometry and flight conditions to construct a dynamic model based on both explicit equations and by correlating related variables using historical data across a vast aircraft database. The AAA geometric model is shown in Fig. 2. The second modeling software used in this project was Athena Vortex Lattice (AVL), developed by MIT. This method applies an extended vortex lattice method to the aircraft lifting surfaces at a specified flow condition to determine forces and moments. The AVL geometric model is shown in Fig. 2. Flight testing was conducted on the single aircraft case, and the dynamic model was fine-tuned by making slight adjustments to stability and control derivatives until the recorded flight data and the physics based model converged across large portions of the flight. Additional flight testing was then conducted to validate the improved dynamic model. The notation for these stability and control derivatives are developed and outlined in reference [23].

IV. GUIDANCE LOGIC

The guidance algorithms are used to compute the UAS state commands required to transition the system from the current to the desired position. In this research, a moving point guidance algorithm is implemented that can be used for any arbitrary trajectory curve. This method computes the necessary speed ($V_T$), roll angle ($\phi$), pitch angle ($\theta$) and sideslip angle ($\beta$) commands. The moving points indicate the desired position and velocity in 3-dimensional space in an inertial coordinate frame depicted by the following equation:

$$o[k] : \begin{cases} 
\rho^o[k] = p_N^o[k], \rho_E^o[k], \rho_H^o[k] \\
v^o[k] = v_N^o[k], v_E^o[k], v_H^o[k] 
\end{cases}$$  (9)
This is a discrete time ($k$) equation where $o[k]$ indicates the moving point array consisting of a position and a velocity vector in inertial NEH frame, which are essentially different from vehicles position and velocity vectors. For longitudinal error, the following equation is used to compute $\theta_{cmd}$.

$$\theta_{cmd} = \tan^{-1}\left( k_{aLon} \cdot \eta_{Lon} \left( k_{pLon} + \frac{k_{iLon}}{s} \right) \right)$$

(10)

$k_{pLon}$ and $k_{iLon}$ are proportional and integral gains for the PI controller. $k_{aLon}$ is the adaptive gain but for this project it is held constant. For lateral error, $\phi_{cmd}$ is computed using lateral acceleration, outlined in Ref. [9].

$$\phi_{cmd} = \tan^{-1}\left( \frac{a_{xcmd}}{g} \right)$$

(11)

To improve the guidance performance, an acceleration feedback signal is used. From Ref. [24], lateral acceleration during real flight test is not always valid for Eq. 11 since the UAS is not always in a level turn. The feedback loop scheme is shown in Fig. 3.

The sideslip angle command, $\beta_{cmd}$ is designated as zero. For each sample time $[k]$, two moving points are used from the current sample time $[k]$ and from $[k-1]$.

$$d = R \cdot \sin(\varepsilon - \tau)$$

(13)

$$\dot{d} = V \cdot \sin(\psi_E - \tau)$$

(14)

where $R$ is the length from waypoint 1 ($p_1$) to a reference point on the UAS center of gravity (c.g.). Angles $\tau$, $\varepsilon$, and $\psi$ are shown in Figure 4. With the assumption of small sideslip angle in the absence of path curvature, the derivative of Eq. 14 can be written as follows:
Now Eq. 15 is linearized to develop an LTI model for the lateral guidance:

\[ \ddot{d} = |V| \dot{\psi}_E \cos(\psi_E - \tau) \]  (15)

The goal of LQR control theory is to minimize the cost function, \( J \) using \( Q \) and \( R \) weighting matrices.

\[ J = \int (x^T Q x + u^T R u) \ dt \]  (19)

Here, \( R \) was assumed to be 1 and \( Q \) is defined as the adaptive matrix. Considering there is no specific control or tracking requirements on the side velocity \( (v_n) \), \( q_2 \) is set to be 1, now by solving algebraic Ricatti Equation \( q_1 \) and \( u = a_n \) can be calculated as follows:

\[ Q = \begin{bmatrix} q_{11} & 0 \\ 0 & q_{22} \end{bmatrix} = \begin{bmatrix} \frac{d_b}{|V|} & 0 \\ 0 & 1 \end{bmatrix} \]  (20)

\[ u_{Lat} = -q_1 \dot{d} + \sqrt{2q_1 + q_2^2 |V|} \]  (21)

where \( d_b \) is the maximum allowed cross-track error.

VI. Validation and Verification Test

In this section, results from the validation and verification testing will be presented.

A. Avionics

The hardware used in this research primarily consists of two processor boards: a Tegra K1 and a custom built Data Acquisition Board (DAQ) ([27]). It also consists of Xbee 900 MHz communication devices with traditional UAS state sensors. State sensors include GPS (Global Positioning System), IMU (Inertial Measurement Unit) and dynamic pressure sensor. For GPS and IMU, a VectorNAV VN-200 module is used. An AMS 5812 pressure sensor is used for measuring the dynamic pressure (air-speed) via a pitot tube.

The Tegra K1 is a powerful processor containing four ARM Cortex-A15 cores running at 2.3 GHz, and has 192 Kepler based GPU cores. The VectorNAV sensor and the DAQ board are connected to this board. The GNC algorithm runs as a ROS node on Tegra K1.

The DAQ board acts as an interface for data exchange between the sensors and the Tegra. The board is connected to all sensors except for GPS and IMU. It collects data from these sensors and sends it to Tegra through serial communication. The DAQ board also receives the controller outputs from Tegra and sends them to the servos. This board houses an Arduino Mega processor that is its primary computational platform.

A telemetry radio device Xbee Pro 900 XB9B-DMST-002 is used as the main communication transceiver for exchanging data between the ground control station and between both aircraft. Communication reliability and robustness are the most important factors for maintaining formation in cooperative aircraft flight. The Xbee supports mesh networking that enables efficient and fast data packet transfers to and from multiple units in real time. Experiments show that the maximum round trip time taken by the two aircraft in exchanging data is 2 milliseconds.

The aircraft uses its current states from sensor information, user-defined waypoints in the flight area, and the position and velocity of potential obstacles in order to plan a trajectory in real time. Control commands are output from the Tegra and executed via the aircraft engine and control surface servos.

In this experiment, a 2nd aircraft was placed in the track of the flying aircraft as a stationary obstacle to be avoided using morphing potential methods.
B. Software and Hardware in the Loop

Prior to the flight testing, system testing was conducted in both the software-in-the-loop (SiTL) and hardware-in-the-loop (HiTL) environments. The SiTL testing involved simulations within the Matlab Simulink workspace on a desktop computer that generates aircraft states, feeds them into the flight controller, and executes the control output of the system in order to determine the aircraft states at the following iteration. This SiTL testing was done to ensure that the coding itself was correct and no logic errors could be found through extensive situational testing. Following the SiTL testing, the flight controller software was transferred to the embedded processor on the aircraft. During this HiTL testing, the aircraft 6-DoF equations of motion were propagated in the Matlab Simulink environment and were sent through serial communication to both aircraft. The flying aircraft used these states and the positional information from the stationary aircraft to determine its corresponding trajectory and control outputs. Both aircraft simultaneously broadcasted their information to the other, along with the ground station computer. This testing was done to ensure that the software was implemented correctly aboard the aircraft avionics and significantly improved flight testing efficiency and reduced the risks involved in flight testing operations. The system architecture for HiTL testing is shown in Figure 5, and the physical setup can be seen in Figure 7.

Results from the HiTL testing for the systems is shown in Figure 8, where it can be clearly seen that the flying aircraft avoids the stationary aircraft, and returns to its desired trajectory following the avoidance maneuver.

A HiTL simulation scenario where two aircraft flying head-to-head trajectory, following the same four waypoints, but in opposite directions, is used to further validate the effectiveness of morphing potential field. In this case each aircraft communicates through Xbee modules, with each other and the ground station. The position and velocity
information are shared consistently amongst each aircraft in order to formulate a real-time morphing potential field. Each aircraft has its own decentralized collision avoidance algorithm running onboard that allows to create an adaptive potential field around each aircraft. The results are shown in Figure 9.

U1 and U2 represent the respective aircraft, with U1 flying clockwise direction and U2 flying counter clockwise. It can be clearly seen that the aircraft avoid each other efficiently at each South-East (first encounter) and North-West (second encounter) corners.

C. Flight Test Setting

Flight testing was conducted at the Clinton International Model Airport in Lawrence, Kansas. System settings for the aircraft were identical to the HiTL setup, except for the aircraft states being read by the onboard sensors instead of through the Matlab SimuLink serial interface. Both aircraft were powered on, and one aircraft was placed in the center of the flight path for the waypoints. A pilot takes off the aircraft from ground using a remote control (RC) and continues to control it through climb phase, and once at the desired altitude and in a trimmed, steady level-wing flight, the autopilot was engaged. Using onboard sensor information, user-defined waypoints, and the real-time position of the grounded aircraft, the flying aircraft navigated around the desired waypoints while avoiding the grounded aircraft.

D. Flight Test Results

Flight testing performed on the aircraft systems produced similar results to those seen from the HiTL testing, namely that the flying aircraft autonomously tracks the waypoints while avoiding the grounded aircraft in the 2D plane. In this test, the position of a stationary aircraft was used in real-time for successful morphing potential field-based collision avoidance on an airborne platform. A simulation was created using the initial conditions for the autopilot from the flight test, and these simulation results were compared to the flight testing results for validation of the 6DoF aircraft dynamic models used in this experiment. These results are shown in the following figures. Observable differences between the flight test and the simulation can be attributed to a number of factors, such as variations between the true and estimated dynamic models, noise characteristics from sensors due to engine-induced structural vibrations, and external forces such as wind gusts. However, despite these factors the system was able to produce the desired performance in an unstructured environment.
Figure 6 represents four subplots indicating 2-D trajectory tracking North-East view (top left), East-Height view (top right), rate of change of altitude (bottom left) and 3-D trajectory tracking (bottom right). In each of these sub-figures, data from flight test is compared with 6-DoF simulations' data. In the 2-D trajectory tracking plot, at around $t = 186\, s$ time stamp, the overhead flying aircraft starts avoiding the grounded aircraft by entering a roll left maneuver. This is clearly comparable in performance and avoidance radius to the HiTL simulation results in figure 8. Due to asymmetric potential field around the grounded aircraft, it can be clearly observed from the flight test result trajectory, that the flying aircraft returned to its original path quite efficiently and quickly.

Figures 10 and 11 represent the longitudinal and lateral-directional states respectively, in addition to control surface outputs. In subplot of figure 10, it can be observed that the velocity tracking is within $10\, ft/s$ in spite of collision avoidance maneuvers which could potentially make the tracking inaccurate. This further shows the effectiveness of the morphing concept in the collision avoidance algorithm, that makes the avoidance relatively smooth and energy efficient.

In both the figures 10 and 11, the state and control surface time trajectories are well matched with the trajectories in simulations, indicating a robust GNC system and accuracy in the physics based dynamic model. These results were achieved despite high wind conditions ($4 - 6$) knots. The avoidance maneuvers occurred around the 90 and 200 second marks of the autonomous flight in Figure 11 and can be observed by the large bank angles of the aircraft.

VII. Conclusion

A novel automatic collision avoidance algorithm that morphs itself using the relative velocity and approach velocity was developed and successfully verified and validated using a large unmanned aerial system. Several validation and verification flight tests were conducted using a DG-808 UAS with a 4-meter wingspan flying at 35 knots and a fixed beacon on the ground. The DG-808 UAS was instrumented using an advanced in-house autopilot system and was controlled using an LQR optimal flight controller. By only knowing the position and velocity of the obstacle, the DG-808 UAS successfully and autonomously departed from the predefined trajectory that was designed by the flight test team avoiding collision with the fixed obstacle. The UAS efficiently returned to its desired path after the avoidance maneuver and continued flying the normal trajectory. No human interaction was required and only the GPS position and velocity of platforms were shared between the aircraft and obstacle over a meshed network.

Acknowledgment

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.
REFERENCES