



Adaptive Control of Precision Guided Airdrop Systems with Highly Uncertain Dynamics

Martin R. Cacan* and Mark Costello†

Georgia Institute of Technology, Atlanta, Georgia 30322

DOI: 10.2514/1.G003039

The bulk of research in the field of precision guided airdrop systems has focused on improving landing accuracy in the presence of atmospheric winds that can exceed vehicle airspeed. One important challenge of parafoil systems is their highly uncertain flight dynamic behavior and control response, which can result from canopy degradation or an offnominal inflation event. This significantly impacts the ability to reach the target and can often lead to very large miss distances. This work addresses guided airdrop system model uncertainty with a novel combined direct and indirect adaptive control strategy to quickly characterize vehicle dynamics and lateral control sensitivity in flight. Extensive simulation and experimental flight testing indicate that the proposed adaptive algorithm is capable of high-accuracy landing in a large variety of degraded conditions, including unknown nonlinear changes in control sensitivity as well as control reversals. In comparison, current industry standard algorithms experience over an order of magnitude decrease in accuracy when tested under identical scenarios.

Nomenclature

$f()$	=	static nonlinear function of Hammerstein model
$G(z)$	=	transfer function of Hammerstein model
u, v, y	=	Hammerstein model input; scaled intermediate control value; and model output
V_{wx}, V_{wy}	=	horizontal components of the atmospheric wind
V_0	=	vehicle airspeed
x_p, y_p, z_p	=	inertial position of the vehicle with respect to the target
α, β	=	parameters of $G(z)$
γ	=	Hammerstein model scaling parameter
δ_a	=	nondimensional asymmetric brake input
η, b	=	linear gain and y offset of approximate static nonlinear function
$\mu()$	=	mean value of $()$
$\psi, \dot{\psi}$	=	inertial heading and heading rate
$\hat{()}$	=	estimate of $()$

I. Introduction

PRECISION guided airdrop systems use sensor feedback and onboard actuators in conjunction with guidance, navigation, and control (GNC) algorithms to actively steer the aerial vehicle to a desired impact point (IP). This enables higher release altitudes and improved landing accuracy in comparison to ballistic (unguided) parachutes. Initial research made use of ram-air inflatable parafoils and a beacon signal to home toward the IP [1–4]. When military-grade Global Positioning System (GPS) signal quality was made available to public and private enterprises, more advanced GNC schemes were developed, given that position, velocity, and time data were available for feedback control [5–14].

GNC algorithms are designed to control mathematically derived dynamic models that predict actual flight characteristics of the

system. Development of these autonomous algorithms requires extensive system identification in which entire flights are conducted with open-loop commands chosen to characterize aerodynamic parameters and control sensitivity mappings. However, parafoil canopies are severely prone to changes in flight characteristics due to the very fact that they are high flexible and redeployable systems. Any deviation from the expected performance increasingly inhibits the ability of the control algorithm to steer the system to the target accurately.

To date, relatively little work has been conducted to address the large-scale deviations parafoil systems can exemplify, such as the torn canopies shown in Fig. 1. The advantages of in-flight system identification, particularly vehicle airspeed, were shown to improve path planning and wind estimation [13,15–17]. Most adaptive parafoil systems have been directed at coping with variable payload mass and linear estimation of the control sensitivity [10,14,18,19]. A significantly larger body of research exists, studying adaptive control methodologies for fixed-wing aircraft [20–22]. However, these methods are often poorly suited for application to guided airdrop systems due to relatively short flight durations to learn system parameters, heavy susceptibility to atmospheric winds, and limited sensor feedback.

This work presents and analyzes an in-flight system identification scheme capable of learning the vehicle dynamics between deployment and landing using a data-driven modeling approach. The vehicle model identification can be prioritized early when there is excess altitude and exploited at the end of the flight for accurate landing. Guided airdrop systems are well matched to this approach because the majority of the flight can be spent loitering near the target waiting for landing. This period can be modified to explore the control sensitivity mappings and exploit the behavior during the final stages of flight when accuracy is most important. The proposed method represents the vehicle dynamics using a Hammerstein model that represents the nonlinear system as a static nonlinear scaling block and linear time-invariant dynamics. This model captures the general characteristics of a precision airdrop system in a simple and concise manner, which can be realized rapidly with minimal computational effort.

This work leverages and modifies a Hammerstein model identification method originally proposed by Bai [23] to create a novel adaptive control strategy for hardware implementation on a guided airdrop system. This dual direct and indirect adaptive control strategy significantly expands the robustness to failure modes of parafoil systems in comparison to current literature. The details of the in-flight system identification algorithm of a Hammerstein model are presented in Sec. II. This method is integrated into a GNC algorithm in Sec. III for robust landing capabilities for a highly uncertain dynamic vehicle. Section IV provides a description of the simulation model and

Presented as Paper 2017-3722 at the Aerodynamic Decelerator Conference, Denver, CO, 5–9 June 2017; received 9 May 2017; revision received 29 November 2017; accepted for publication 3 December 2017; published online 28 March 2018. Copyright © 2018 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the ISSN 0731-5090 (print) or 1533-3884 (online) to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*Graduate Research Assistant, Woodruff School of Mechanical Engineering, Center for Advanced Machine Mobility.

†David S. Lewis Professor of Autonomy, Guggenheim School of Aerospace Engineering and Woodruff School of Mechanical Engineering, Center for Advanced Machine Mobility.

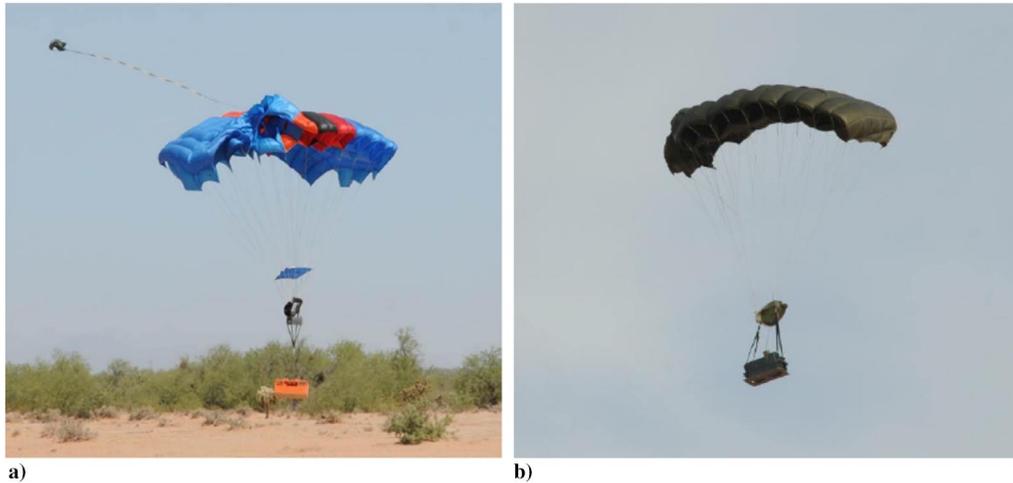


Fig. 1 Precision airdrop system with ripped a) upper-surface canopy and b) right canopy cells caused by high shock loading during inflation [19].

experimental vehicle used to generate results presented in Sec. V. Conclusions and major findings of the work are discussed in Sec. VI.

II. In-Flight System Identification

In many typical scenarios, the majority of the flight time for a guided airdrop system is spent homing to and loitering near the target until the system approaches the ground. This time can be used to characterize the vehicle system dynamics and control sensitivity. Fortunately, a significant tracking error can be tolerated at high altitudes when the system behavior is being learned by the controller. When the vehicle reaches low altitude, the learned model can provide high tracking accuracy for precision landing. Additionally, feedback for guided airdrop systems is typically limited to GPS only, which places the focus on position control via steady maneuvers due to limited attitude feedback and velocity control authority. This allows the system identification algorithm to prioritize characterization of the steady-state control sensitivity between a lateral input command and the associated turn rate response.

Flight time, and hence identification time, is capped based on the release altitude. To accomplish in-flight system identification, a system model is needed that has sufficient complexity to capture the dynamic behavior but has a simple enough form to ensure model convergence over a short period of flight time. For parafoil systems, the turn rate dynamics can be approximated as linear with minimal error [24]. However, the control sensitivity is often nonlinear and can exhibit significant deviations due to damage. The distinct linear and nonlinear characteristics of a parafoil system are well represented by the Hammerstein model. The Hammerstein model is a specific type of nonlinear model that splits the plant into two elements: a linear time-invariant dynamic block, and a static (or memoryless) nonlinear block. Figure 2 shows an illustration of the model where an input vector $\bar{u}(t)$ can be scaled or mixed by the static nonlinear mapping (SNLM) to generate an intermediate, nonlinear control parameter $\bar{v}(t)$. The intermediate input vector excites the linear time-invariant dynamics (LTID) to produce the system response $\bar{y}(t)$. As a result, the goal of the identification process is to characterize the SNLM function $f(\bar{u})$ and the LTID transfer function matrix $\bar{G}(z)$.

For sufficiently short model identification time, the dynamic behavior and control sensitivity of the vehicle can be characterized in flight before final approach of the vehicle to the IP for a fully adaptive control algorithm.

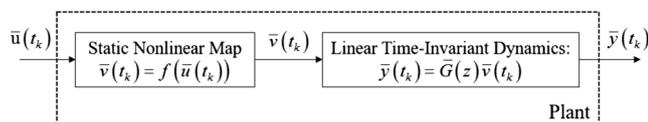


Fig. 2 Graphic representation of a Hammerstein model.

Many identification methods for Hammerstein models have been suggested in the literature [25–32]. This work applies the separable least-squares method proposed by Bai [23] and expanded to the multiple-input/multiple-output (MIMO) case by Jeng and Huang [33]. This method separates the identification of the LTID and SNLM blocks, detailed in Secs. II.A and II.B, respectively. This allows the dynamics to be realized quickly after canopy inflation, and the remainder of the flight is devoted to learning the nonlinear control mapping. The details of this method are presented in the following for the single-input/single-output (SISO) case, although the method can be extended to the MIMO case with proper accounting for cross-channel effects [24,33]. For generality, the goal is to characterize the relationship between an input command u and system response y .

A. Identification of Dynamics

The key to separating the two elements of the Hammerstein model is through the use of a pseudorandom binary sequence (PRBS) or a square wave of changing frequency [23]. A PRBS signal switches between two values $\pm c$ for some $c \neq 0$ in a deterministic, although uncorrelated, fashion. As a result of only being excited by two input levels, the SNLM acts as a static gain, which is visualized in Fig. 3. During the identification of the dynamics, the true, nonlinear control mapping is approximated by a linear function with constant gain and bias. In fact, the linear gain of the SNLM and the gain of the LTID transfer function are coupled because the system $(\gamma G(z), f(u)/\gamma)$ is identical for all nonzero values of a scaling factor γ .

During dynamic identification, the linear function PRBS(c) commands the input between $\pm c$, causing the control input to be a square wave and the intermediate nonlinear control to be characterized by linear approximation:

$$u(k) = \text{PRBS}(c) \quad (1)$$

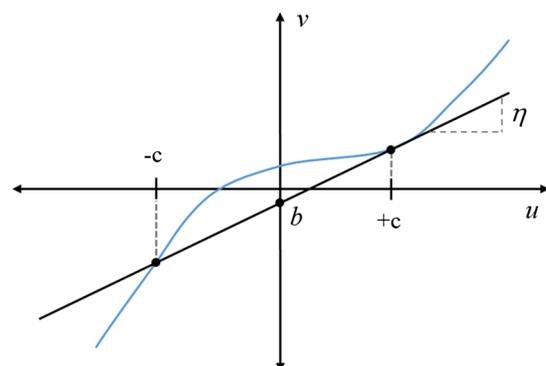


Fig. 3 Linear approximation of an example SNLM under PRBS input.

$$v(k) = f(u(k)) \approx f(\text{PRBS}(c)) = \pm \eta c + b \quad (2)$$

Note that, with additional math, the PRBS value c need not be symmetrical about zero. As a result of the nonuniqueness of the Hammerstein model, the linear channel sensitivity η is assumed equal to one and the scaling parameter γ is chosen autonomously after realization of $G(z)$ to select a specific solution.

With the PRBS input command, the model response can be defined as follows:

$$y(k) = G(z)v(k) \quad (3)$$

$$= G(z)u(k) + G(z)b \quad (4)$$

where

$$G(z) = \frac{\alpha_1 z^{-1} + \dots + \alpha_n z^{-n}}{1 + \beta_1 z^{-1} + \dots + \beta_n z^{-n}} \quad (5)$$

The order of the linear dynamics is assumed to be on the order of n for generality but must be selected by an engineer in advance. After expanding the differential equation corresponding to Eqs. (3–5), and solving for the output, it is clear that the constant term b only adds steady-state offset to the system outputs B :

$$y(k) = - \sum_{s=1}^n \beta_s y(k-s) + \sum_{s=1}^n \alpha_s u(k-s) + b \sum_{s=1}^n \alpha_s \quad (6)$$

$$y(k) = - \sum_{s=1}^n \beta_s y(k-s) + \sum_{s=1}^n \alpha_s u(k-s) + B \quad (7)$$

This can be rewritten in vector form using the data and parameter vectors: $\phi(k)$ and θ , respectively. In this form, the parameter vector can be solved using a least-squares approach, done in real time through an extended Kalman filter:

$$y(k) = \phi^T(k)\theta \quad (8)$$

$$\phi(k) = \{u(k-1), \dots, u(k-n), -y(k-1), \dots, -y(k-n), 1\}^T \quad (9)$$

$$\theta = \{\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n, B\}^T \quad (10)$$

The estimated input–output relationship $\hat{y}(k) = \phi^T(k)\hat{\theta}$, given by $\hat{G}(z)$, can be calculated after sufficient PRBS excitation. Additionally, the parameter vector is initialized based on the nominal performance of the vehicle. The nonuniqueness issue of Hammerstein models is addressed by selecting the value of the scaling parameter γ such that the gain of the transfer function matrix is unity, pushing all of the linear scaling into the SNLM:

$$\gamma = \frac{\sum_{s=1}^n \hat{\alpha}_s}{1 + \sum_{s=1}^n \hat{\beta}_s} \quad (11)$$

When $\hat{G}(z)$ has unit gain, the steady-state model response directly tracks the intermediate input. This implies that the intermediate control parameter represents the desired steady-state response and the SNLM itself is the control effectiveness from an input command to steady-state output. At the conclusion of the dynamic identification process, the LTID block and a linear estimate of the SNLM block are known:

$$v(k) = f(u(k)) \approx \gamma \frac{B}{\sum_{s=1}^n \alpha_s} + \gamma u(k) \quad (12)$$

B. Identification of Steady-State Control Sensitivity

The goal of this section is to improve upon the initial linear estimate of the SNLM calculated in Eq. (12) by using real-time input and output data. This approach does not assume a structure or parametric model to the SNLM block, which provides the controller with flexibility to capture damaged, nonlinear control sensitivities, including asymmetric behavior and dead bands.

Given that the SNLM characterizes control input to steady-state output, estimation could be conducted by time averaging a series of constant turns (similar to typical open-loop system identification). This method estimates the control mapping accurately but very slowly makes it infeasible for this application. A more efficient way estimates the intermediate control signal using the inverse of estimated dynamics [23,32]. The intermediate control can be estimated by simply using the inverse of the dynamics: $\hat{v}(k) = \hat{G}(z)^{-1}y(k)$. This enables a regression on a series of points $(u(k), \hat{v}(k))$ for $k = 1, \dots, N$ to estimate $f()$ in either a parametric or nonparametric fashion. However, without a guarantee, the identified dynamics are minimum phase and the inverse dynamics may be unstable, prohibiting the estimation of $\hat{v}(k)$. Additionally, an inverted dynamic model has lead-compensator characteristics that amplify the measurement noise of the output signal.

Another option is available by reframing the problem. The SNLM represents a static function from commanded input to desired steady-state output. This time-invariant function can pass through the linear dynamic model without distortion because the LTID has unity gain. This identifies the nonlinear mapping associated with the Wiener model (where the dynamic block precedes the nonlinear function) and is valid for SISO systems and MIMO systems that are diagonally dominant. Previously, $f()$ was estimated by comparing values (u, v) from Eq. (13). With this modification, $G(z)$ acts as the input first and $f()$ can be estimated by studying the data (u_F, y) , where $u_F(k) = G(z)u(k)$ as in Eq. (14). Figure 4 depicts how this new method combines the two signals to directly compare parameters at a single instance of time without having dynamic inversion:

$$y = G(z)f(u) \rightarrow G^{-1}(z)y = v = f(u) \quad (13)$$

$$y = f(G(z)u) \rightarrow y = f(u_F) \quad (14)$$

One of the primary benefits of this method is that the SNLM can be averaged in the parameter domain instead of temporally. Time averaging is a slow process commonly conducted in flight identification procedures. Instead, this method logs all data points $(u_F(k), y(k))$ into the time-invariant parameter space (u_F, y) by accounting for the dynamic (temporal) response of the system in u_F . This is shown in the SNLM estimate block in Fig. 4, where points are added to this control authority space. Noise is removed by fitting any parametric or nonparametric model to estimate the control sensitivity $f()$. Further clarification of the method is provided in Sec. III.

III. Adaptive Guidance, Navigation, and Control

At a basic level, the guidance algorithm conducts path planning based on current state estimates. The navigation algorithm uses GPS measurements to estimate vehicle position, velocity, heading direction,

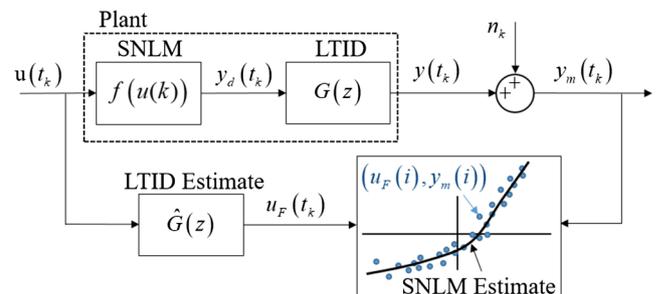


Fig. 4 Representation of the method used to estimate the static nonlinear element of the Hammerstein model.

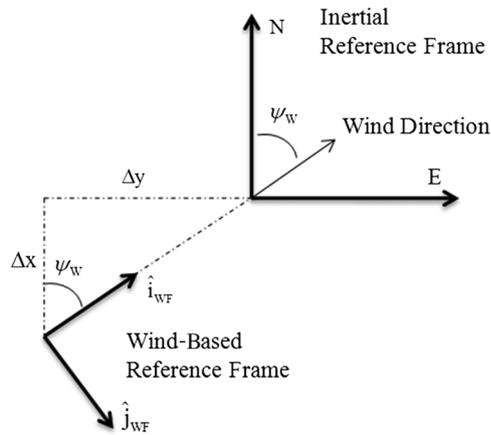


Fig. 5 Visual representation of the wind-based reference frame.

and rate in addition to atmospheric winds. The control algorithm acts upon current state parameters and desired parameters provided by guidance to steer the vehicle along a desired path. The model estimation and learning of the control sensitivity developed in the last section are used to improve these algorithms for increased robustness to damage.

For the lateral-only GNC developed here, the assumed form of the dynamic transfer function that governs the turn rate response of the Hammerstein model is chosen to be first order ($n = 1$). An analysis into higher-order models is difficult due to the level of sensor noise because it characteristically leads to overfit data or systems dominated by the first-order pole. Additionally, it is noted that adding a delay of one regular time update to the first-order model reduces the rms error between the model and the measured output:

$$G(z) = \frac{\alpha z^{-2}}{1 - \beta z^{-1}} \quad (15)$$

A. Guidance

The guidance algorithm uses vehicle state and atmospheric estimates from the navigation algorithm to compute a set of desired paths to accurately reach the IP. To aid path planning, a wind-based reference frame (WF) is established (Fig. 5), which is associated with the drift expected from the influence of the horizontal components of the atmospheric winds [3,15,34]. Using the wind-based reference frame is advantageous because it decouples the absolute system movement into no-wind flight conditions and the drift caused by atmospheric winds.

The expected wind drift (Δx , Δy) is based upon the integral of the wind profile and the descent rate from the ground level to the payload altitude z_p . This is approximated using the estimated mean wind field (\bar{V}_{Wx} , \bar{V}_{Wy}) and the mean descent rate \dot{z}_p :

$$\Delta x = \frac{\bar{V}_{Wx} z_p}{\dot{z}_p}; \quad \Delta y = \frac{\bar{V}_{Wy} z_p}{\dot{z}_p} \quad (16)$$

The orientation of the wind-based reference frame is rotated to align the \hat{i}_{WF} component with the horizontal wind direction: $\psi_W = \tan^{-1}(V_{Wy}, V_{Wx})$. As a result, guidance planning in this frame can neglect the influence of the winds because the frame directly accounts for it.

The path planning is broken into four stages, each with a specific goal. First, the initialization phase executes a series of open-loop commands to estimate the linear dynamics (LTID) of the Hammerstein model, the horizontal wind vector, and vehicle airspeed. Second, the loiter phase maintains proximity near the target and learns the SNLM control sensitivity mapping. Below an altitude threshold, the approach phase uses the learned control mapping to steer the vehicle accurately to the target. Finally, the last stage flares the canopy just before landing to bleed off forward velocity and prevent the payload from tumbling after impact.

1. Initialization

The initialization phase commands the PRBS signal to the trailing-edge brakes, causing the vehicle to execute a series of s turns. The first turn is held until the vehicle executes over a 180 deg turn, during which time Eq. (17) can be used to estimate the horizontal atmospheric wind field using GPS data [17]:

$$\begin{bmatrix} \dot{x}_{P,1}^m - \mu \dot{x}_p & \dot{y}_{P,1}^m - \mu \dot{y}_p \\ \vdots & \vdots \\ \dot{x}_{P,N}^m - \mu \dot{x}_p & \dot{y}_{P,N}^m - \mu \dot{y}_p \end{bmatrix} \begin{Bmatrix} V_{Wx} \\ V_{Wy} \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} (V_1^m)^2 - \mu v^2 \\ \vdots \\ (V_N^m)^2 - \mu v^2 \end{bmatrix} \quad (17)$$

Here, $(\dot{x}_{P,i}^m, \dot{y}_{P,i}^m)$ represent the measured horizontal velocity components, and $(V_i^m)^2$ is the squared velocity magnitude in the set $i = 1, \dots, N$. Finally, the airspeed of the parafoil and payload system V_0 can be estimated using measured GPS data and horizontal wind estimates:

$$V_{0,i} = \sqrt{(\dot{x}_i^m - V_{Wx})^2 + (\dot{y}_i^m - V_{Wy})^2}; \quad V_0 = \mu v_0 \quad (18)$$

Selecting the correct magnitude of the constant input of the PRBS signal is challenging because the vehicle can be very sensitive or more unresponsive. The initial turn rate is chosen to be conservative in the case of a highly sensitive turn rate response and slowly updated based on filtered gyroscope data, which provide an independent turn rate measurement. Equation (17) does not depend on the direction of the turn, however, so only the turn rate magnitude is important. When the magnitude of the turn rate is between 10 and 25 deg/s, the input is held constant and the atmospheric wind estimation process begins. These bounds are selected to ensure sufficient lateral excitation but ensure the system does not enter spiral flight.

After the first s turn of the PRBS is used to conduct the initialization algorithm, the navigation algorithm can be initialized using the estimated vehicle airspeed and atmospheric wind estimates. This defines the vehicle in the wind-based reference frame, and the remaining s turns are modulated in length to fly in the general direction of the target using a homing controller. This is not always effective given the damage to the canopy but, in many degraded and healthy conditions, this method ensures the vehicle maintains proximity to the target while conducting system identification. At the end of the initialization phase, the LTID estimation completes and the system transitions to the loiter phase.

2. Loiter

During this phase, the parafoil and payload system flies through a set of figure-eight turns in order to maintain a position relative to the target. This is accomplished by defining waypoints that are at the tips of a T-shaped reference line that extends downwind of the IP in the wind-based reference frame. The T approach was proposed by Jann [15] and is shown in Fig. 6. The vehicle uses Dubins path planning to reach the desired waypoints from its current location and heading. Dubins paths minimize the flight distance with three maneuver elements: initially turning at a constant maximum rate in the direction of the next waypoint; flying straight to approach the waypoint; and, when near the waypoint, turning to match the desired heading defined by the waypoint [15,35]. Four potential paths exist: two of which are shown in Fig. 6. This method is computationally efficient and only requires accurate knowledge of the control mapping over a subset of the control space based on the three Dubins maneuvers (in the regions around $\dot{\psi} = 0, \pm \dot{\psi}_{\max}$). With a poor estimate of the control mapping early in the loiter phase, the ability to track these paths is degraded. However, as the identification algorithm improves the estimate, the SNLM is well characterized for improved path tracking. Additionally, Dubins pathing can be expanded to include the transition between straight flight and the maximum turn rate by using the settling time of the identified dynamics to account for very slow systems [24].

During the entire loitering period, the altitude required to reach the target from the current location is computed. This is calculated in

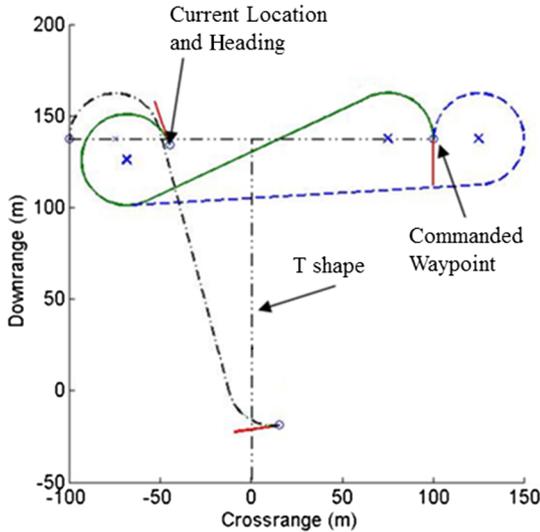


Fig. 6 Visualization of Dubins path planning transitioning from the left waypoint to the right.

Eq. (19) using the Dubins path length to the target L_{Dubins} and the estimated glideslope, V_0/\dot{z}_p :

$$h_{\text{REQ}} = L_{\text{Dubins}} \frac{\dot{z}_p}{V_0} \quad (19)$$

When the current altitude equals h_{REQ} , the system switches into the approach phase.

3. Approach and Flare

The vehicle enters approach when the guidance algorithm calculates that it will reach the target at the same time the system lands. The vehicle deviates from the figure-eight holding pattern and attempts to fly upwind toward the target along the stem of the T. Based on the nature of the WF, this ensures the vehicle is approaching the target facing into the wind to reduce ground speed. When the vehicle is several meters above ground level, the vehicle enters the final stage where the vehicle is commanded to enter straight and level flight. Just before impact, full symmetric brakes are applied to flare the canopy into prestall conditions, which further decreases the forward airspeed and minimizes the potential for the payload to roll after ground impact.

B. Navigation

For guided parafoil and payload systems, the important states to estimate are the atmospheric winds and vehicle position, the velocity, the heading angle, and the rate ($x, y, z, \psi, \dot{\psi}, V_0, V_{W_x}, V_{W_y}$). After the open-loop initialization procedure [Eqs. (17) and (18)] generates an initial estimate of these parameters, the navigation algorithm uses the

GPS to propagate them forward in time using an extended Kalman filter. The details are excluded here for brevity but have been published previously by Cacan et al. [24,36].

C. Control

This adaptive controller improves over conventional controllers that depend on a priori control mappings with an algorithm that focuses on the exploration (during loiter) and exploitation (during approach) of the Hammerstein model. Using the data-driven approach presented in Sec. II.B, the relationship between the asymmetric brake control input and turn rate response can be estimated. The SNLM directly estimates the control mapping of the vehicle, which is approximated using a dynamic lookup table to minimize the constraints on the form of the SNLM estimate. This is represented graphically by a continuous, piecewise linear function.

An illustration of how the SNLM updates due to exploration of a new control space is depicted in Fig. 7. Immediately after Initialization, the SNLM estimate is the linear approximation as shown in Fig. 7a. If, as shown, the guidance algorithm commands a right turn at a maximum turn rate, the controller will use the linear approximation to select the nondimensional input δa_c , which actually induces a greater turn rate response $\dot{\psi}_m$. The estimator uses the data points ($\delta a_c, \dot{\psi}_m$) to update the SNLM, causing the control level to decrease until the measured value equals the commanded, as shown in Fig. 7b. The entire control space can be explored and learned quickly because the guidance strategy selects from three path options (turn left, right, or straight), which significantly reduces the control space that has to be well characterized. Further details on this identification are presented in Sec. V.

With continuously improving estimates of the SNLM, the common control strategy for Hammerstein systems is to cancel out the effects of the SNLM and control the linear dynamic system $G(z)$ [37–39]. This strategy is presented in Fig. 8 and implemented here to track the desired heading rate that is commanded from the guidance algorithm. However, it is important to note that the SNLM estimate cannot blindly be inverted to generate a valid control mapping. It is assumed that the turn rate is a function of the input (in the sense that there is one output for a given input), which enables parameterization of the mapping, but the inverse is not guaranteed due to potential nonlinear behavior. This is resolved by using a map handler that places a dead band across any local nonmonotonic behavior so that the inverse mapping is a proper function. This causes a discontinuity in the control sensitivity block f_{MH}^{-1} , such that no values in this range will be commanded to δa_c . This simply and effectively creates a globally monotonic mapping, although turn rate errors (hence pathing errors) can occur when the actuator is required to pass through these ranges. An analysis indicated that this deadband behavior only significantly impacted landing accuracy when the deadpan crossed $\dot{\psi} = 0$. Additional information is available in [24].

IV. Experimental Vehicle and Simulation Model

Figure 9a shows the experimental vehicle in flight. This small-scale remote-controlled vehicle was designed to provide an efficient

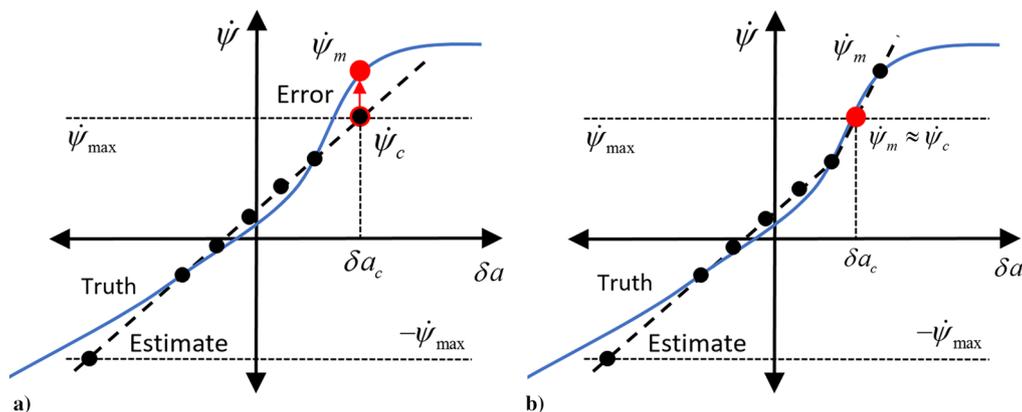


Fig. 7 Illustration of how SNLM updates when exploring a new portion of the control space, given that $t_{(a)} < t_{(b)}$.

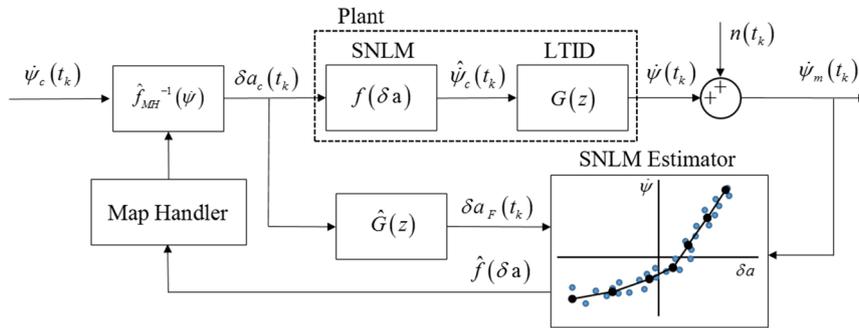


Fig. 8 Block diagram representation of the control strategy used in the hyperadaptive GNC.

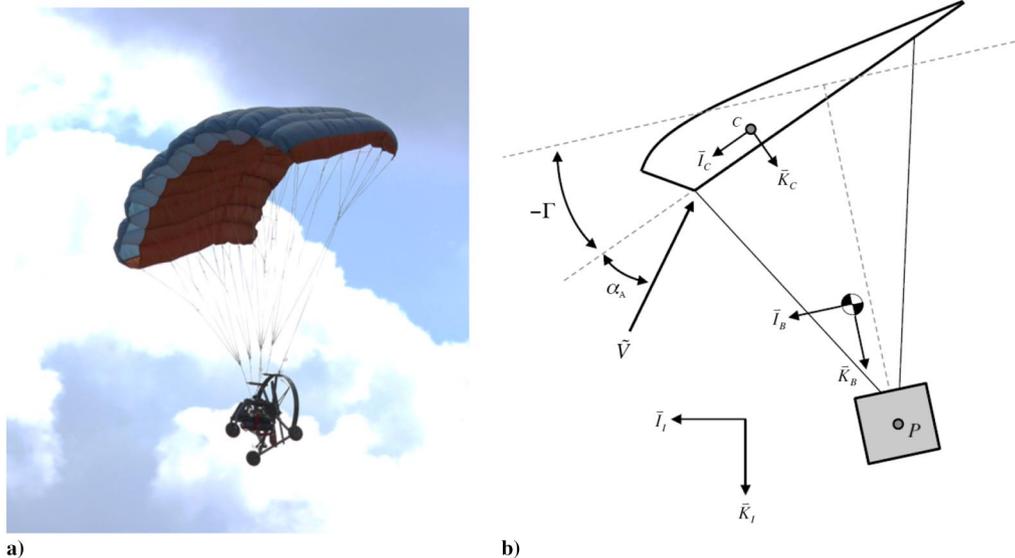


Fig. 9 Precision airdrop system a) in flight and b) as a schematic for dynamic modeling.

alternative to full-size airdrop system testing, which required guided payloads to be dropped from manned aircraft. The 1.35 m² parafoil canopy supported a 2.7 kg payload equipped with an autopilot, a GPS antenna, servo motors, an electric brushless motor, a speed controller, and a battery. Additionally, a small coin cell-powered wireless microelectromechanical system inertial measurement unit was located in the canopy to provide heading rate feedback to insure the initial response under the PRBS signal was within response bounds (only the course rate was available from GPS directly). The autopilot used a PIC32MX-family microcontroller with a clock speed of 80 MHz to handle wireless communications (4 Hz), sensor measurements (16 Hz), and GNC calculations (4 Hz). The vehicle was actuated via two high-torque servo motors, which enabled lateral control of the vehicle through trailing-edge deflection of the canopy.

Figure 9b depicts a schematic of a parafoil and payload system used for dynamic modeling of the vehicle. With the exception of movable parafoil brakes, the parafoil canopy is considered to be a fixed shape. The combined system of the parafoil canopy and the payload are represented by a six-degree-of-freedom rigid-body model, defined by three inertial position components of the total system mass center as well as the three Euler orientation angles. The canopy aerodynamic forces and moments are computed using the wind relative velocity \vec{V} at the canopy aerodynamic center (point C in Fig. 9b). The transformation from the body frame (frame B in Fig. 9b) located at the vehicle center of mass to the canopy reference frame (frame C in Fig. 9b) is defined by a single axis rotation in pitch by the canopy incidence angle Γ . The equations of motion for this six-degree-of-freedom parafoil and payload representation have been derived previously and validated through flight testing [17,24,34,40,41].

Atmospheric winds are captured in the simulation model through low-frequency altitude-dependent variations and high-frequency

turbulence, as shown in Fig. 10. Low-frequency variation is used to introduce horizontal wind shears, which are commonly observed in experimental testing and can significantly degrade landing accuracy [42]. High-frequency wind gusts are captured using a discrete implementation of the Dryden turbulence model [43,44]. In a simple and concise manner, this wind model captures the nature of atmospheric wind fields close to the ground. By statistically varying the upper and lower air mass velocities, the shear altitude, and the gust noise, a rich variety of physical scenarios can be constructed.

V. Results

The performance and capabilities of the proposed adaptive GNC are presented for both simulation and experimental results. As a comparison, two additional GNCs that represent industry standard airdrop GNC strategies are considered. Both use identical navigation and guidance algorithms, except that the initialization phase ends after wind estimates are computed because the LTID identification is removed. The control algorithms, however, are significantly less adaptive. The conventional algorithm uses a proportional-integral (PI) controller to steer the vehicle and requires a priori knowledge of the control sensitivity mapping [13,34]. The integral component is used to estimate and reject steady-state turn bias. In total, this GNC can estimate and adapt to the atmospheric winds, the vehicle airspeed, and the turn rate biases. The second algorithm uses a model reference adaptive controller (MRAC), which can estimate both the linear control sensitivity and bias. The user-defined reference model is assumed to take the following form:

$$\delta a_c = f(\dot{\psi}_c) \approx \delta a_{\text{Gain}} \dot{\psi}_c + \delta a_{\text{bias}} \quad (20)$$

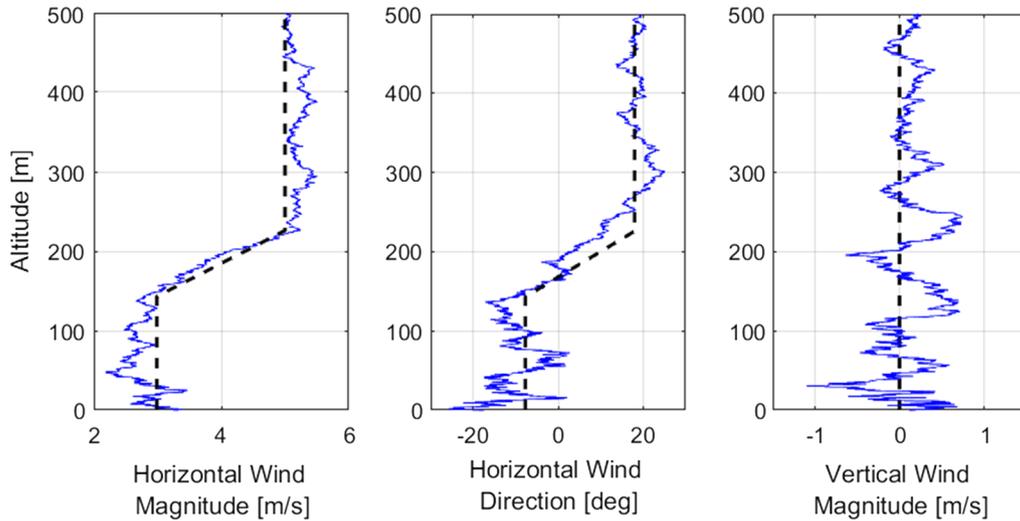


Fig. 10 Simulated wind field showing the base model (dashed) and included atmospheric turbulence (solid).

where δa_{Gain} and δa_{bias} are the control gain and bias, respectively. These parameters are updated at every time step based on the error between the measured turn rate $\dot{\psi}_m$ and the desired turn rate based on the reference model $\dot{\psi}_d$:

$$\frac{d\delta a_{\text{Gain}}}{dt} = -\gamma_1 \dot{\psi}_c (\dot{\psi}_m - \dot{\psi}_d) \quad (21)$$

$$\frac{d\delta a_{\text{bias}}}{dt} = -\gamma_2 (\dot{\psi}_m - \dot{\psi}_d) \quad (22)$$

Here, the γ terms are tuning parameters to be selected offline. To clarify between the flight controllers, the proposed adaptive

algorithm is termed the hyperadaptive GNC and is compared against the conventional and MRAC algorithms.

A. Simulated Example Autonomous Flight

To understand the capabilities of each algorithm, an example damaged flight condition is tested in a simulation environment where the control sensitivity to turn left is greatly reduced. This damage case generally characterizes degraded conditions that only significantly impact one side of the canopy, such as actuator malfunction or canopy damage.

The guided airdrop system is released from a 500 m altitude and is located upwind of the target, following standard industry practice as

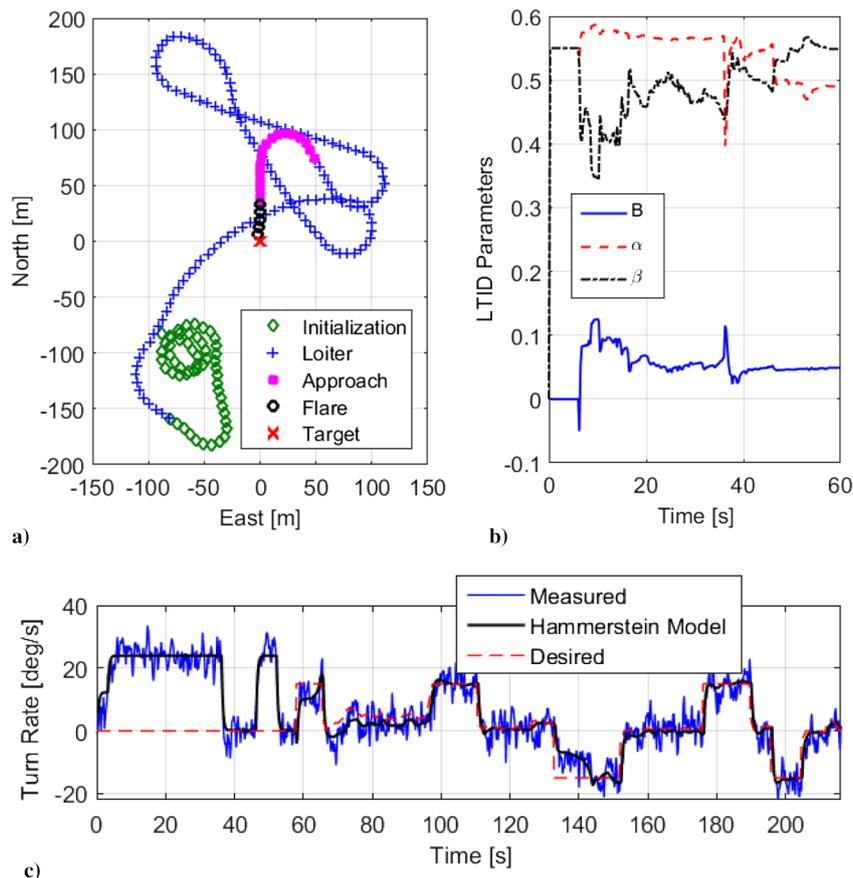


Fig. 11 Simulated a) horizontal trajectory, b) LTID parameter estimate during initialization, and c) turn rate time history of the hyperadaptive GNC steering a vehicle with asymmetric turn rate response.

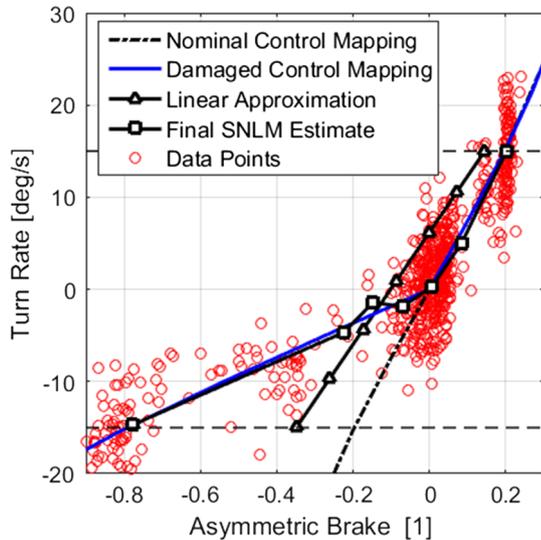


Fig. 12 Initial and final SNLM estimates for a vehicle with asymmetric turn rate response.

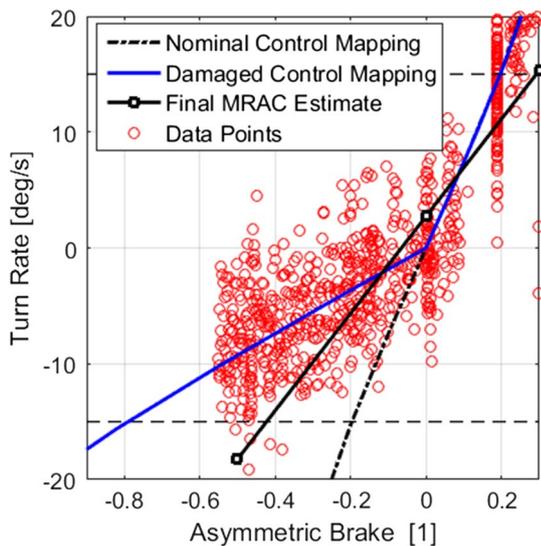


Fig. 13 Control mapping estimated by the MRAC GNC.

shown in Fig. 11a. From $t = 0 \rightarrow 6$ s, the initial turn rate response to the commanded input is corrected to be within bounds (it steps up once). At this point, the initialization and LTID algorithms both begin. Using the initialization algorithms, the airspeed of 7.2 m/s is

estimated with less than a 5% error and the LTID estimation algorithm accurately characterizes the lateral turn rate dynamics during the PRBS input sequence from $t = 6$ s to $t = 58$ s. Before $t = 36$ s, the single control point does not provide sufficient observability to estimate parameters, causing their values to drift as seen in Fig. 11b. After the second, third, and fourth turns, the system is sufficiently excited to estimate the key LTID parameter β to within 2%. The values of the linear mapping accurately approximate the damaged control mapping at the PRBS command points, as can be seen in Fig. 12. Additionally, the final estimate of the Hammerstein model is plotted in Fig. 11c and shows close agreement with the measured response.

After initialization, the system enters loiter and conducts the figure 8 loiter pattern to maintain proximity to the target (seen slanted in Fig. 11a due to the wind). The time period from approximately $t = 60$ s to $t = 130$ s is used to update the SNLM on the side of positive control authority. This shifts the estimate of the SNLM down from the linear estimate to align with the measured response. When the vehicle first commands a maximum rate left turn at $t \approx 130$ s, a significant tracking error populates the SNLM estimator with points in the region of $(\delta a, \psi) = (-0.4, -10)$ in Fig. 12. The SNLM estimate is able to update to match the true turn rate sensitivity in approximately 10 s. At the conclusion of the flight, the final SNLM estimate shows a strong match to the true damaged control mapping. The ability to track the desired turn rate after $t \approx 170$ s enables the system to land less than 10 m from the target.

In comparison, the results of the MRAC estimation of the same control sensitivity mapping are presented in Fig. 13. The linear approximation of the asymmetric control mapping results in slow turns to the left and a turn that is too sharp to the right. The inability to track the desired turn rates defined by the Dubins paths causes the system to land over 90 m from the target. Finally, the conventional GNC also has significant problems handling this type of error. The PI controller can only account for the turn rate error by varying the turn rate bias to match only the current point. After a short series of turns, the control increases, causing the system to enter an undesirable spiral flight.

B. Simulated Monte Carlo Analysis

The three GNC algorithms are stress tested by randomly varying a set of parameters that alters flight performance of the vehicle. This includes the canopy incidence angle, which controls airspeed and descent rate; asymmetric scaling, dead bands, and control reversals of the control mapping; an actuator time constant, which changes the settling time of the turn rate dynamics; and atmospheric winds. Landing dispersions for the conventional, MRAC, and hyperadaptive algorithms controlling a highly uncertain precision payload system are presented in Fig. 14. Conventional, MRAC, and hyperadaptive simulation results indicate the algorithms have 50% CEPs of 74.6, 34.5, and 27.9 m, respectively, which show a threefold improvement in the hyperadaptive GNC over the conventional GNC. Table 1 presents the landing statistics for both the nominal and highly

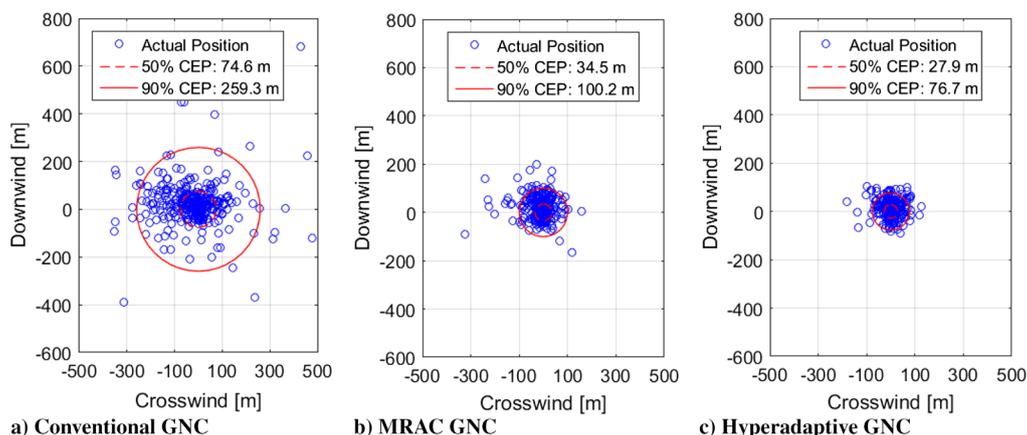


Fig. 14 Monte Carlo landing dispersions of three adaptive GNC algorithms tested on a vehicle with highly uncertain dynamics.

Table 1 Landing statistics for the multi-parameter Monte Carlo simulation

GNC algorithm	50% CEP	Change vs nominal	90% CEP	Change vs nominal
Conventional (nominal)	19.2 m	—	50.9 m	—
MRAC (nominal)	17.4 m	—	51.9 m	—
Hyperadaptive (nominal)	19.8 m	—	57.7 m	—
Conventional (damaged)	74.6 m	↑ 288%	163.59 m	↑ 410%
MRAC (damaged)	34.5 m	↑ 98%	100.2 m	↑ 93%
Hyperadaptive (damaged)	27.9 m	↑ 41%	76.7 m	↑ 33%

uncertain vehicle configurations in the presence of realistic wind fields. The conventional GNC experiences a 288% rise in 50% circular error probable (CEP) and an over 400% increase in 90% CEP when the uncertainty is introduced to the vehicle. The linear turn rate sensitivity adaption of the MRAC GNC reduces the effect of uncertainty but still exhibits a 100% decrease in landing accuracy for both landing metrics. Finally, the hyperadaptive GNC is the most robust to the model uncertainty, with increases of only 41 and 33% for the 50 and 90% CEPs, respectively.

Not included in the previous Monte Carlo analysis is the case of control reversal. Most adaptive control algorithms including the MRAC controller assume the sign of the control authority is known to update parameters in the correct direction. However, the flexible nature of the proposed hyperadaptive algorithm allows for the identification of a control reversal during the dynamics estimation. If the dynamic response is flipped from the input, the gain of the transfer function is negative, which causes the linear approximation of the SNLM to also be reversed. When the previous study is repeated with only the control reversal and atmospheric winds, the hyperadaptive algorithm is able to maintain nominal landing accuracy. The conventional and MRAC GNC algorithms, however, experience 2500 and 900% increases, respectively, over their nominal accuracy due to destabilization of the controller.

C. Experimental Flight Tests

The final validation of the adaptive GNC was through experimental flight testing in a real-world environment. In addition to the nominal vehicle configuration, software modifications external to the GNC were used to implement a control reversal, asymmetric control sensitivity, and decreased actuator speed. These tests changed both the control sensitivity mapping and the turn rate dynamics to accurately gauge the algorithm's ability to handle a large range of model uncertainty. A fifth test loaded identical software onto a second small-scale parafoil and payload system. This second vehicle had a different payload mass and shortened servo actuation arms to

Table 2 Experimental flight test results for the hyperadaptive algorithm

Vehicle configuration	Number of flights	50% CEP	90% CEP
Nominal	11	21.2 m	66.3 m
Control reversal	11	44.4 m	55.8 m
Asymmetric control sensitivity	9	42.4 m	63.4 m
Slow actuators	11	59.5 m	90.0 m
Second vehicle	9	21.5 m	43.0 m
Combined	51	35.9 m	69.0 m

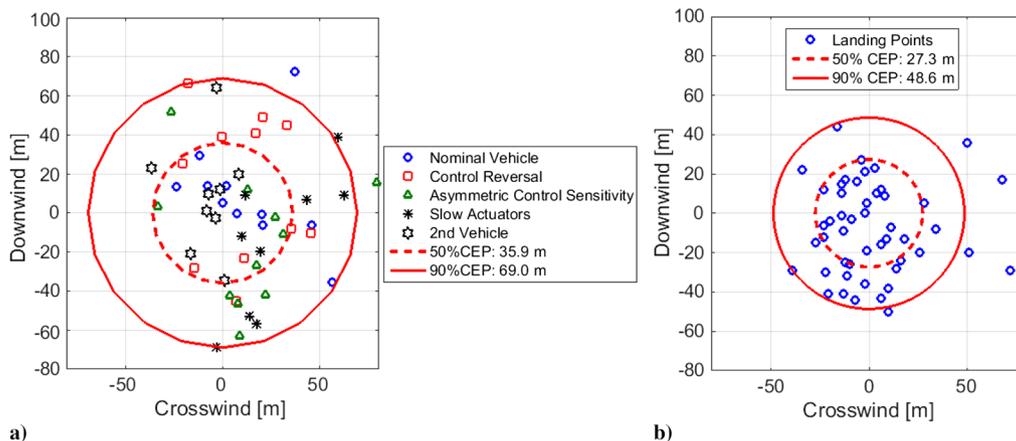
reduce the control sensitivity, and the canopy was smaller and poorly rigged, resulting in a strong turn bias. The conventional algorithm was flown concurrently, but only on the nominal system as a baseline, and to account for variable atmospheric conditions.

Flight testing was conducted south of Atlanta, Georgia in a large wooded clearing across all times of the day to test the algorithms in typically calm morning winds and more turbulent afternoon conditions. The results of 51 flights across all case studies are presented in Fig. 15a, with individual landing metrics listed in Table 2. Aggregate results of the hyperadaptive GNC indicate a 50% CEP of 35.9 m and a 90% CEP of 69.0 m. This aligns well with the multiparameter simulation testing, which has 50 and 90% CEPs of 27.9 and 76.7 m, respectively. Part of the variation seen between landing results in Table 2 is due to relatively few sample points not fully capturing the full range of atmospheric wind fields. However, the slow actuator case is expected to degrade performance because the rate limiting of the actuator makes the vehicle less maneuverable, and hence less agile to reject error.

As a reference, the conventional GNC controlling only the nominal system has 50 and 90% CEPs of 27.3 and 48.6 m, respectively. The hyperadaptive algorithm is capable of achieving nearly equivalent landing accuracy in the face of large variations in vehicle flight performance. Under similar circumstances, the conventional control algorithm would have damaged the payload and experienced significant increase in landing error, as shown by the simulation results.

VI. Conclusions

An adaptive guidance, navigation, and control algorithm for precision guided airdrop systems was developed to overcome highly uncertain flight characteristics. At its core was a model identification algorithm that fully characterized the vehicle turn rate dynamics and control sensitivity mapping during the course of a single flight. A data-driven nonparametric approach to modeling the control sensitivity was used to handle a wide array of damage. Extensive simulation results and flight-test data showed that the proposed adaptive GNC was robust to all forms of model uncertainty commonly experienced in practice. Minimal to no loss of landing accuracy was noted for changes in atmospheric winds, vehicle airspeed, turn rate biases, actuator speed, and many linear and nonlinear changes to the control sensitivity mapping, including nonmonotonic and deadband behaviors. As a

**Fig. 15** Experimental landing results of the a) hyperadaptive GNC and b) conventional GNC algorithms.

comparison, two GNC algorithms based on currently fielded flight software were studied. Both algorithms performed equivalently for small levels of damage, including changes to the vehicle airspeed or turn rate biases. However, nonlinear control sensitivity and control reversals posed a significant error and caused over an order of magnitude decrease in landing accuracy.

Acknowledgment

The authors would like to acknowledge the support of the Natick Soldier Research Development and Engineering Center Airdrop Technology Team.

References

- [1] Barton, W., and Knapp, C., "Controlled Recovery of Payloads at Large Glide Distances, Using the Para-Foil," *Journal of Aircraft*, Vol. 5, No. 2, 1968, pp. 112–118.
doi:10.2514/3.43917
- [2] Goodrick, T., "Estimation of Wind Effect on Gliding Parachute Cargo Systems Using Computer Simulation," *AIAA Aerodynamic Decelerator Systems Conference*, AIAA Paper 1970-1193, 1970, pp. 14–16.
- [3] Goodrick, T. F., and Murphy, A., "Analysis of Various Automatic Homing Techniques for Gliding Airdrop Systems with Comparative Performance in Adverse Winds," *4th AIAA Aerodynamic Deceleration Systems Conference*, AIAA Paper 1973-0462, 1973.
- [4] Goodrick, T., "Hardware Options for Gliding Airdrop Guidance Systems," *6th AIAA Aerodynamic Decelerator and Balloon Technology Conference*, AIAA Paper 1979-0471, 1979.
- [5] Wright, R., Benney, R., and McHugh, J., "Precision Airdrop System," *18th AIAA Aerodynamic Decelerator Systems Technology Conference*, AIAA Paper 2005-1644, 2005.
- [6] Benney, R., Barber, J., McGrath, J., McHugh, J., Noetscher, G., and Tavan, S., "The Joint Precision Airdrop System Advanced Concept Technology Demonstration," *18th AIAA Aerodynamic Decelerator Systems Technology Conference*, AIAA Paper 2005-1601, 2005.
- [7] Bennett, T., and Fox, R., "Design, Development and Flight Testing of the NASA X-38 7500 ft² Parafoil Recovery System," *17th AIAA Aerodynamic Decelerator Systems Technology Conference*, AIAA Paper 2003-2107, 2003.
- [8] Hogue, J. R., and Jex, H. R., "Applying Parachute Canopy Control and Guidance Methodology to Advanced Precision Airborne Delivery Systems," *AIAA Aerodynamic Decelerator Systems Technology Conference*, AIAA Paper 1995-1537, 1995, pp. 15–18.
- [9] Murray, J. E., Sim, A. G., Neufeld, D. C., Rennich, P., Norris, S., and Hughes, W., "Further Development and Flight Test of an Autonomous Precision Landing System Using a Parafoil," Office of Management, Scientific and Technical Information Program, NASA TM 4599, July 1994.
- [10] Calise, A., and Preston, D., "Design of a Stability Augmentation System for Airdrop of Autonomous Guided Parafoils," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2006-6776, 2006.
- [11] Calise, A. J., and Preston, D., "Swarming/Flocking and Collision Avoidance for Mass Airdrop of Autonomous Guided Parafoils," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 1123–1132.
doi:10.2514/1.28586
- [12] Slegers, N., and Costello, M., "Aspects of Control for a Parafoil and Payload System," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, 2003, pp. 898–905.
doi:10.2514/2.6933
- [13] Ward, M., and Costello, M., "Adaptive Glide Slope Control for Parafoil and Payload Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 1019–1034.
doi:10.2514/1.59260
- [14] Ward, M., Cacan, M., Scheuermann, E., Costello, M., Bergeron, K., Noetscher, G., and Shurtliff, M., "Flight Test Results of Recent Advances in Precision Airdrop Guidance, Navigation, and Control Logic," *23rd AIAA Aerodynamic Decelerator Systems Technology Conference*, AIAA Paper 2015-2148, 2015.
- [15] Jann, T., "Advanced Features for Autonomous Parafoil Guidance, Navigation and Control," *Aerodynamic Decelerator Systems Technology Conferences*, AIAA Paper 2005-1642, 2005.
doi:10.2514/6.2005-1642
- [16] Carter, D., George, S., Hattis, P., McConley, M., Rasmussen, S., Singh, L., and Tavan, S., "Autonomous Large Parafoil Guidance, Navigation, and Control System Design Status," AIAA Paper 2007-2514, 2007.
- [17] Ward, M., Costello, M., and Slegers, N., "Specialized System Identification for Parafoil and Payload Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 2, 2012, pp. 588–597.
doi:10.2514/1.53364
- [18] Calise, A. J., Yang, B.-J., and Craig, J. I., "Augmenting Adaptive Approach to Control of Flexible Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, 2004, pp. 387–396.
doi:10.2514/1.2441
- [19] Culpepper, S., Ward, M., and Costello, M., "Adaptive Control of Damaged Parafoils," *22nd AIAA Aerodynamic Decelerator Systems Technology Conference*, AIAA Paper 2013-1344, 2013.
- [20] Calise, A. J., Hovakimyan, N., and Idan, M., "Adaptive Output Feedback Control of Nonlinear Systems Using Neural Networks," *Automatica*, Vol. 37, No. 8, 2001, pp. 1201–1211.
doi:10.1016/S0005-1098(01)00070-X
- [21] Morelli, E. A., "Real-Time Global Nonlinear Aerodynamic Modeling for Learn-to-Fly," *AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2016-2010, 2016.
- [22] Hwang, I., Kim, S., Kim, Y., and Seah, C. E., "A Survey of Fault Detection, Isolation, and Reconfiguration Methods," *IEEE Transactions on Control Systems Technology*, Vol. 18, No. 3, 2010, pp. 636–653.
doi:10.1109/TCST.2009.2026285
- [23] Bai, E.-W., "Decoupling the Linear and Nonlinear Parts in Hammerstein Model Identification," *Automatica*, Vol. 40, No. 4, 2004, pp. 671–676.
doi:10.1016/j.automatica.2003.11.007
- [24] Cacan, M. R., "Adaptive Control of Autonomous Airdrop Systems in Degraded Conditions," Ph.D. Thesis, Georgia Inst. of Technology, Atlanta, GA, 2016.
- [25] Narendra, K., and Gallman, P., "An Iterative Method for the Identification of Nonlinear Systems Using a Hammerstein Model," *IEEE Transactions on Automatic Control*, Vol. 11, No. 3, 1966, pp. 546–550.
doi:10.1109/TAC.1966.1098387
- [26] Billings, S., and Fakhouri, S., "Identification of Systems Containing Linear Dynamic and Static Nonlinear Elements," *Automatica*, Vol. 18, No. 1, 1982, pp. 15–26.
doi:10.1016/0005-1098(82)90022-X
- [27] Pawlak, M., "On the Series Expansion Approach to the Identification of Hammerstein Systems," *IEEE Transactions on Automatic Control*, Vol. 36, No. 6, 1991, pp. 763–767.
doi:10.1109/9.86954
- [28] Bai, E.-W., "An Optimal Two-Stage Identification Algorithm for Hammerstein–Wiener Nonlinear Systems," *Automatica*, Vol. 34, No. 3, 1998, pp. 333–338.
doi:10.1016/S0005-1098(97)00198-2
- [29] Boutayeb, M., Aubry, D., and Darouach, M., "A Robust and Recursive Identification Method for MISO Hammerstein Model," *UKACC International Conference on Control '96*, Vol. 1, IET, 1996, pp. 234–239.
- [30] Chang, F., and Luus, R., "A Noniterative Method for Identification Using Hammerstein Model," *IEEE Transactions on Automatic Control*, Vol. 16, No. 5, 1971, pp. 464–468.
doi:10.1109/TAC.1971.1099787
- [31] Hsia, T., "A Multi-Stage Least Squares Method for Identifying Hammerstein Model Nonlinear Systems," *1976 IEEE Conference on Decision and Control Including the 15th Symposium on Adaptive Processes*, IEEE Publ., Piscataway, NJ, 1976, pp. 934–938.
- [32] Ding, F., and Chen, T., "Identification of Hammerstein Nonlinear ARMAX Systems," *Automatica*, Vol. 41, No. 9, 2005, pp. 1479–1489.
doi:10.1016/j.automatica.2005.03.026
- [33] Jeng, J.-C., and Huang, H.-P., "Nonparametric Identification for Control of MIMO Hammerstein Systems," *Industrial and Engineering Chemistry Research*, Vol. 47, No. 17, 2008, pp. 6640–6647.
doi:10.1021/ie071512q
- [34] Ward, M., "Adaptive Glide Slope Control for Parafoil and Payload Aircraft," Ph.D. Thesis, Georgia Inst. of Technology, Atlanta, GA, 2012.
- [35] Dubins, L. E., "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, Vol. 79, No. 3, 1957, pp. 497–516.
doi:10.2307/2372560
- [36] Cacan, M. R., Scheuermann, E., Ward, M., Costello, M., and Slegers, N., "Autonomous Airdrop Systems Employing Ground Wind Measurements for Improved Landing Accuracy," *IEEE/ASME Transactions on Mechatronics*, Vol. 20, No. 6, Dec. 2015, pp. 3060–3070.
- [37] Bloemen, H., Van Den Boom, T., and Verbruggen, H., "Model-Based Predictive Control for Hammerstein-Wiener Systems," *International*

- Journal of Control*, Vol. 74, No. 5, 2001, pp. 482–495.
doi:10.1080/00207170010014061
- [38] Fruzzetti, K., Palazolu, A., and McDonald, K., “Nonlinear Model Predictive Control Using Hammerstein Models,” *Journal of Process Control*, Vol. 7, No. 1, 1997, pp. 31–41.
doi:10.1016/S0959-1524(97)80001-B
- [39] Norquay, S. J., Palazoglu, A., and Romagnoli, J., “Model Predictive Control Based on Wiener Models,” *Chemical Engineering Science*, Vol. 53, No. 1, 1998, pp. 75–84.
doi:10.1016/S0009-2509(97)00195-4
- [40] Ward, M., Costello, M., and Slegers, N., “On the Benefits of In-Flight System Identification for Autonomous Airdrop Systems,” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, 2010, pp. 1313–1326.
doi:10.2514/1.49030
- [41] Scheuermann, E., “Autonomous Control of Parafoil and Payload Systems Using Upper Surface Canopy Spoilers,” Ph.D. Thesis, Georgia Inst. of Technology, Atlanta, GA, 2015.
- [42] Yakimenko, O., Slegers, N., and Tiaden, R., “Development and Testing of the Miniature Aerial Delivery System Snowflake,” AIAA Paper 2009-2980, 2009.
- [43] “Flying Qualities of Piloted Aircraft,” U.S. Dept. of Defense MIL-STD-1797A, 2004.
- [44] Gage, S., “Creating a Unified Graphical Wind Turbulence Model from Multiple Specifications,” *AIAA Modeling and Simulation Technologies Conference and Exhibit*, AIAA Paper 2003-5529, 2003.