JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 36, No. 5, September–October 2013

Robust Parafoil Terminal Guidance Using Massively Parallel Processing

Jonathan Rogers*
Texas A&M University, College Station, Texas 77843
and
Nathan Slegers†
University of Alabama in Huntsville, Huntsville, Alabama 35899

DOI: 10.2514/1.59782

Terminal guidance of autonomous parafoils is a difficult problem in which wind uncertainty and system underactuation are major challenges. Existing strategies almost exclusively use impact error as the criterion for optimality. Practical airdrop systems, however, must also include other criteria that may be even more important than impact error for some missions, such as ground speed at impact or constraints imposed by drop zones with restrictions on flight patterns. Furthermore, existing guidance schemes determine terminal trajectories using deterministic wind information and may result in a solution that works in ideal wind but may be sensitive to variations. The work described here develops a guidance strategy that uses massively parallel Monte Carlo simulation performed on a graphics processing unit to rank candidate trajectories in terms of robustness to wind uncertainty. The result is robust guidance, as opposed to optimal guidance. Through simulation results, the proposed path planning scheme proves more robust in realistic dynamic wind environments compared with previous optimal trajectory planners that assume perfect knowledge of a constant wind.

Nomenclature

B	=	discrete model control sensitivity
D	=	distance of the turn initial point with respect to the target
$\mathbf{i}_T$, $\mathbf{j}_T$, $\mathbf{k}_T$	=	target frame unit vectors
L	=	distance to target along target line
R	=	final turn radius
$\mathbf{s}$	=	terminal parafoil state vector
T_{app}, $T_{\text{app}}^{\text{des}}$	=	final approach time and desired final approach time
t_{pre}	=	final turn advance timing
t_0	=	time final turn begins
t_1	=	time final approach begins
t_2	=	time of predicted impact
V_h, V_v	=	parafoil horizontal and vertical speed
W_x, W_y	=	target frame wind speed components
x, y, z	=	parafoil inertial positions in the target frame
Δ	=	predictive controller discrete sampling period
Δt	=	final turn time
δ_a	=	parafoil asymmetric brake deflection
$\boldsymbol{\sigma}$	=	impact statistics vector from graphics processing unit
τ	=	turning time constant
ψ	=	parafoil heading
ψ_F	=	final approach angle

I. Introduction

GUIDED parafoil systems provide a unique capability for precision airdrop that is highly attractive for military resupply missions. Specifically, parafoils enable accurate payload delivery at considerable standoff distances, reducing risk to cargo aircraft. Currently, the U.S. military is interested in developing autonomous guided parafoils in an array of sizes, from very small designs weighing tens of kilograms to large vehicles weighing up to 14,000 kg [1]. These next-generation autonomous parafoils are expected to provide unprecedented landing accuracy, even in unfavorable environmental conditions, using advanced sensor technology and optimized control algorithms. Although parafoils, as a class of unmanned systems, are fundamentally simple to control due to their stability and benign dynamics, several factors make the specific task of precision airdrop extremely challenging. First, low airspeed and the absence of thrust make parafoils extremely sensitive to atmospheric winds, which can be highly unpredictable at low altitudes. Second, parafoils are underactuated systems in that they typically employ only lateral control through asymmetric brake deflection, with longitudinal control for guidance being extremely limited. Although longitudinal control is limited for ram-air parachutes, a useable amount of glide slope control becomes available as the system lands into the wind due to the control in airspeed available with symmetric brake deflection [2,3]. Winds are often a primary driver in landing point accuracy, and it is a common occurrence that wind speed approaches the vehicle airspeed such that advancing upwind is difficult.

Over the last two decades, significant advances have been made in the capability of airdrop systems with many guided systems converging on similar guidance strategies and now reliably achieving accuracies of 250 ft [4,5]. These improvements have come from improvements in mission planning [6] that provide release point prediction, the ability to incorporate up-to-date atmospheric conditions by dropsondes, and from improving guidance techniques. Although numerous guidance strategies for parafoils have been developed, most are essentially composed of an energy management phase, a homing phase, and a terminal guidance phase [4]. After the parafoil is released upwind of the target, energy management begins by establishing a loiter pattern in which the parafoil dissipates altitude. The homing phase comprises the transition from energy management to the terminal guidance phase and occurs at an altitude allowing the parafoil to proceed directly to the target area. Terminal guidance consists of final maneuvers such that the parafoil impacts the target, usually flying upwind so that its groundspeed is at a minimum. Descriptions of different terminal guidance strategies, their benefits and drawbacks, and advanced techniques based on skydivers are provided in [2]. Terminal guidance is crucial in that it is

Presented as Paper 2012-4736 at the AIAA Atmospheric Flight Mechanics Conference, Minneapolis, MN, 13–16 August 2012; received 26 July 2012; revision received 26 November 2012; accepted for publication 8 December 2012; published online 17 June 2013.

*Assistant Professor, Department of Aerospace Engineering.
†Associate Professor, School of Mechanical and Aerospace Engineering.

quite difficult to plan an approach path such that the parafoil arrives at the target and the ground simultaneously while flying upwind; consequently, it is in terminal guidance where most existing algorithms differ. Although energy management and homing are necessary, to limit the scope of the problem, this paper assumes homing and energy management have been successfully completed so as to bring the parafoil to a heading, altitude, and distance to the target such that terminal guidance can begin.

Simple parafoil guidance strategies have been proposed [4,7] in which energy management consists of a spiral of varying radius over the target area, and terminal guidance is a straight line path to the target. Such trajectories cannot guarantee landing into the wind, and Calise and Preston [8] showed that they are largely not suitable for high wind-to-airspeed ratios. Several other guidance schemes have been proposed [9–13] such as optimal control, trajectory databases, model predictive control (MPC), and direct glide slope control. More recently, terminal guidance strategies have been investigated through generation of optimal trajectory paths. Carter et al. [14] proposed a bandwidth-limited trajectory planner in which five parameters were optimized to minimize final position and heading errors. Rademacher et al. [15] developed a hybrid method in which optimal terminal trajectories are derived as either modified Dubins paths or minimum-control trajectories with multiple optimization parameters. Slegers and Yakimenko [16] recently developed a terminal guidance planning scheme using the direct method of the calculus of variations and inverse dynamics in a virtual domain. By constraining the trajectory to a specific functional form, the problem is reduced to a single parameter optimization problem and can be solved very efficiently. A comparison of the terminal path planners proposed in [14–16] demonstrates how variation in the assumed shape of the trajectory and requirements of the controller can lead to extremely different guidance structures while still achieving similar results.

Although precision placement guidance has advanced, a limitation that still exists is that most of the strategies exclusively emphasize touchdown error as the criterion for optimality. Error is clearly a concern; however, practical airdrop systems must also include other criteria that may be as important as or even more important than error. As an example, impact sensitive payloads such as liquids or blood in humanitarian drops require low impact velocities. In these cases, decreasing impact velocity at the cost of increased error may be a more optimal solution. Similarly, space-limited or complex drop zones may impose geometric constraints. As examples, geographical features such as canyons and bodies of water may alter the desired dispersion shape. Likewise, in combat scenarios, unsecured areas may impose severe restrictions on the drop zone. Integration of the various additional criteria for optimality required for precision airdrop into advanced algorithms such as those proposed in [14–16] can be formidable and often require a complete reformulation of the problem. Another limitation of existing guidance is that they are based on deterministic knowledge of the wind, and thus are susceptible to significant error if wind conditions are unknown or change during terminal flight. A deterministic solution may be appropriate based on the known mean wind; however, it could be extremely sensitive to variations in the wind, with a small change resulting in potentially large errors. As examples, consider potential missions that have either a limit on maximum impact velocity or a constrained drop zone in which, because of an obstacle, a region is forbidden. Using a deterministic solution, the optimal impact may occur close to the mission boundary, but still be acceptable, and selected as suitable. However, in the presence of uncertain winds, many of the solutions may result in mission failure. In contrast, a probabilistic solution would determine potential trajectory sensitivity to wind variation and as a result select, on average, a solution which reduces the probability of violating mission requirements at the expense of final touchdown error. This illustrates that, because winds can never be mapped precisely, it may be more important to select a terminal trajectory that is robust to adjustments due to imperfect wind knowledge rather than an optimal trajectory assuming perfect knowledge of winds. This tradeoff exists not only for terminal guidance but for energy management and homing as well.

In light of the limitations of existing precision airdrop terminal guidance, this paper develops a fundamentally new strategy to address the need for robust guidance, as opposed to optimal guidance, by exploiting the "embarrassingly parallel" nature of Monte Carlo simulations. Embarrassingly parallel refers to types of problems in which little effort is required to separate the problem into a number of parallel tasks. The terminal guidance planning method proposed here relies on online, massively parallel Monte Carlo simulation to predict impact point variance resulting from unknown or changing wind conditions. The rise of massively parallel computing, specifically using graphics processing units (GPUs), has recently enabled unprecedented reductions in run times in a variety of general purpose computing applications. GPUs have been leveraged across a wide spectrum of computational problems from molecular dynamics [17,18], incompressible flows [19], and *N*-body problems [20] to optical flow processing [21] and computational fluid dynamics [22]. Ilg et al. [23] recently showed that run times of six degree-of-freedom (DOF) Monte Carlo simulations can be reduced by up to an order of magnitude when performed on a GPU, creating the possibility that such simulations can be run efficiently as part of a real-time autonomous guidance system. In fact, embedded processors containing onboard graphics accelerators have recently become available [24] that leverage the same processing constructs as desktop GPUs. These embedded GPUs are rapidly becoming more capable, and are already suited to perform real-time massively parallel computations as part of an onboard guidance system.

This paper develops a new robust terminal trajectory planning algorithm for autonomous parafoils. The trajectory algorithm is designed such that, at initiation of terminal guidance, wind uncertainty is evaluated by computing the variance of wind estimates over the course of flight to that point. A set of candidate trajectory paths are generated assuming perfect knowledge of wind conditions. Then, a GPU Monte Carlo simulation is performed for each candidate trajectory using a 6-DOF parafoil model and wind variation. Simulation results are then analyzed to determine which candidate paths embody the desired level of robustness given the level of uncertainty in wind conditions. This is accomplished through minimization of a cost function that penalizes both impact point accuracy and a variety of available dispersion statistics available from the GPU. Extensive simulations of the robust trajectory planning system demonstrate several strengths of the proposed algorithm, including ease and flexibility in designing mission optimality conditions and the ability to shape the dispersion map probability density using intuitive conditions. It is also shown that the robust solution chosen can often vary significantly from what a mission designer may choose by intuition.

II. Parafoil Terminal Guidance Open-Loop Trajectory Planning

Energy management and homing strategies before terminal guidance are flexible and are only required to bring the parafoil to a final turn initiation point (TIP) with sufficient altitude so that a final terminal guidance stage can reach the target in the current wind environment. Terminal guidance, however, requires precision because of the constraint that, at its completion, the parafoil must simultaneously land upwind and be at the target location. A baseline terminal guidance trajectory is described in Fig. 1, in which $\mathbf{i}_T$ and $\mathbf{j}_T$ are axes of the standard north-east-down target reference frame. Nominally the target reference frame is aligned with the prevailing wind so that the crosswind component W_y s zero. However, temporal and spatial wind variations may result is some nonzero crosswind W_y during terminal guidance.

Before entering the final turn, the parafoil is a distance L upwind of the target and a distance $2R$ to the left of the target, where R is the final turn radius. The final turn is initiated at time t_0 when the TIP is reached at a distance D past the target. After reaching the TIP, the final turn is defined by a commanded $\psi(t)$ with boundary conditions $\psi = 0$ at t_0 and $\psi = \psi_F$ at t_1. The resulting final turn time is then $\Delta t = t_1 - t_0$. The final approach (FA) begins at time t_1 where

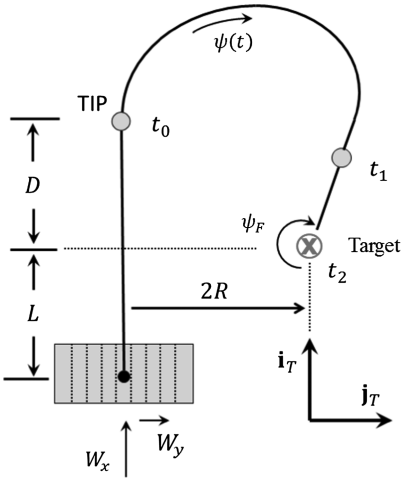

Fig. 1 Terminal guidance geometry.

$\psi = \psi_F$ until impact at t_2. The desired FA time T_{app} is $t_2 - t_1$, where t_2 is at touchdown.

For practical applications, the target location, turn radius R, a desired T_{app}, and ψ_F are specified, whereas the wind components W_x and W_y are estimated online and the altitude z and L are measured. A statement of the terminal guidance problem becomes as follows: For a parafoil at altitude z and a distance L from the target, in the presence of winds W_x and W_y, find the distance D past the target to initiate the final turn trajectory $\psi(t)$ for an ideal impact at t_2 with final direction ψ_F. The general problem takes the form of a two-point boundary value problem for a nonlinear system. However, through two sets of simplifying assumptions, an analytic solution can be found. The first set of assumptions simplifies the parafoil dynamic model by assuming a slow turn rate, so that the roll and sideslip angles can be ignored, a nearly constant descent rate V_v, and constant horizontal airspeed V_h. In this case, parafoil motion reduces to a kinematic model represented by three components of the ground velocity in the target axes

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} = \begin{Bmatrix} W_x + V_h \cos\psi \\ W_y + V_h \sin\psi \\ V_z \end{Bmatrix} \tag{1}$$

where x, y, and z are positions in the target frame. The second simplifying assumption is the form of the commanded turn $\psi(t)$. The form can be chosen as any function that satisfies the boundary conditions $\psi(t_0) = 0$ and $\psi(t_1) = \psi_F$ with respect to the target reference frame. A simple solution is to select a constant turn rate $\dot{\psi}$ so that $\psi = (t - t_0)\dot{\psi}$ during the final turn and $\psi = \psi_F$ during the final approach. Integration of the horizontal velocity from t_0 to t_2 subject to the known constraints results in the following solutions to the final turn time and turn rate:

$$\dot{\psi} = \psi_F / \Delta t \tag{2}$$

$$\Delta t = \frac{\psi_F[2R - (W_y + V_h \sin\psi_F)T_{\text{app}}]}{W_y\psi_F + V_h(1 - \cos\psi_F)} \tag{3}$$

Similarly, integration of velocities along $\mathbf{i}_T$ and $\mathbf{k}_T$ from the current position to t_2 yield

$$D + W_x\Delta t + \frac{V_h}{\dot{\psi}} \sin\psi_F + (W_x + V_h \cos\psi_F)T_{\text{app}} = 0 \tag{4}$$

$$z + V_z\frac{L + D}{W_x + V_h} + V_z\Delta t + V_z T_{\text{app}} = 0 \tag{5}$$

Equations (4) and (5) can then be solved for D and T_{app}, resulting in

$$\begin{aligned} D = {} & -W_x\Delta t - \frac{V_h}{\dot{\psi}} \sin\psi_F \\ & - \frac{(W_x + V_h \cos\psi_F)(W_x + V_h)}{V_h(1 - \cos\psi_F)} \\ & \times \left[\frac{-z}{V_z} - \Delta t - \frac{L - W_x\Delta t - (V_h/\dot{\psi})\sin\psi_F}{W_x + V_h}\right] \end{aligned} \tag{6}$$

$$T_{\text{app}} = \frac{W_x + V_h}{V_h(1 - \cos\psi_F)}\left(\frac{-z}{V_z} - \Delta t\right) - \frac{L - W_x\Delta t - (V_h/\dot{\psi})\sin\psi_F}{V_h(1 - \cos\psi_F)} \tag{7}$$

The guidance strategy then becomes to select a desired approach time $T_{\text{app}}^{\text{des}}$ and final approach direction ψ_F, which are used to evaluate Eq. (7) while loitering in the energy management stage. The altitude at which to begin homing is found by solving Eq. (7) for z, resulting in

$$z = -V_z\left[\Delta t + \frac{V_h(1 - \cos\psi_F)}{W_x + V_h}T_{\text{app}}^{\text{des}} + \frac{L - W_x\Delta t - (V_h/\dot{\psi})\sin\psi_F}{W_x + V_h}\right] \tag{8}$$

Energy management continues while the current altitude is higher than the altitude in Eq. (8), which achieves the desired final approach. Homing is initiated once the altitude decreases to satisfy Eq. (8). During homing, the parafoil approaches the TIP defined by $x = D$ and $y = -2R$, where D is continually estimated during homing using Eqs. (2), (3), (6), and (7) and current estimates of V_v, V_h, W_x, and W_y. Note that, once homing has been entered, neither D nor T_{app} remain free variables. Disturbances and tracking errors while homing will alter the ideal terminal guidance, resulting in T_{app} from Eq. (7) varying from the desired approach time.

The terminal guidance outlined earlier is an open-loop trajectory planner. Once the final turn has begun, the initial conditions and final turn rate from Eqs. (2) and (3) can be used with the kinematic model in Eq. (1) to form the complete x-y and $\psi(t)$ trajectory time histories until impact. To track the identified final turn, any number of trajectory tracking controllers may be used. However, because the terminal guidance prescribes a desired ψ over the terminal trajectory horizon, MPC as described in [3,16] is well suited and easy to implement. In contrast to more complex MPC controllers [11,25,26], a single DOF linear turn model may be used as a plant, demonstrated by Ward [3], to provide good performance. The linear discrete turn model can be represented as

$$\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta \\ 0 & 1 - \Delta/\tau \end{bmatrix}\begin{bmatrix} \dot{\psi} \\ \ddot{\psi} \end{bmatrix}_k + \begin{bmatrix} 0 \\ B\Delta/\tau \end{bmatrix}\delta_{a,k} \tag{9}$$

where ψ and $\dot{\psi}$ are the parafoil heading and turn rates, Δ is the discrete sampling period, B is the control sensitivity, τ is the turning time constant, and δ_a is the parafoil asymmetric brake deflection.

An MPC algorithm, using the previous model, is used for all stages of guidance including homing, the final turn, and the final approach. The ideal terminal guidance in Fig. 1 is based on the simplified kinematic model of Eq. (2) and assumes an ideal transition from zero to constant turn rate once the TIP is reached. In practical systems, the turning dynamics have a nonzero time constant τ. To compensate for turning dynamics, the transition from homing to tracking the final turn is advanced by $t_{\text{pre}}s$, where t_{pre} is a tuning parameter that can be selected based on the turning time constant τ of the particular parafoil system.

A demonstration of the terminal guidance process is shown using a nonlinear 6-DOF parafoil simulation from homing until impact. The 6-DOF model is described in [27] with all model parameters listed in Table 1. The model is numerically integrated using a fourth-order Runge–Kutta algorithm with a time step of 0.05 s. The MPC update rate is 4 Hz and uses $\tau = 2.2$ s, $B = 0.88$ rad/s, a prediction horizon

Table 1 Parafoil Model Parameters

Model parameters	Values
System mass m_B, slug	0.145
Included mass m_I, slug	0.006
Steady-state aerodynamic velocity V_h, ft/s	24.8
Steady-state descent rate V_z, ft/s	13.5
Canopy reference area S, ft^2	10.0
Canopy span b, ft	4.55
Canopy chord $\bar{c}$, ft	2.25
Incidence angle Γ, rad	−0.21
Payload inertia matrix elements, slug · ft^2	$I_{xx} = 0.031$, $I_{yy} = 0.020$, $I_{zz} = 0.040$, $I_{xz} = -0.005$
Canopy inertia matrix elements, slug · ft^2	$I_{xx} = 0.009$, $I_{yy} = 0.006$, $I_{zz} = 0.005$, $I_{xz} = 0$
Elements of the apparent mass matrix, slug	$A = 0.00084$, $B = 0.0022$, $C = 0.029$
Elements of the apparent inertia matrix, slug · ft^2	$P = 0.040$, $Q = 0.010$, $R = 0.0018$
z distance from C to payload mass center z_{CS}, ft	0.3
x distance from C to parafoil mass center x_{CB}, ft	0.5
z distance from C to parafoil mass center z_{CB}, ft	−2.25
x distance from C to canopy rotation x_{CR}, ft	−0.5
z distance from C to canopy rotation z_{CR}, ft	−2.7
x distance from R to canopy aerodynamic center x_{RP}, ft	0.633
x distance from R to apparent mass center x_{RM}, ft	0.59
z distance from R to apparent mass center z_{RM}, ft	0.20
Maximum brake deflection $\bar{d}$, ft	0.75
Aerodynamic coefficients	$C_{D0} = 0.15$, $C_{D\alpha 2} = 0.90$, $C_{Y\beta} = -0.15$, $C_{L0} = 0.25$, $C_{L\alpha} = 0.68$, $C_{m0} = 0.0$, $C_{mq} = -0.265$, $C_{lp} = -0.355$, $C_{l\delta a} = -0.0003$, $C_{nr} = -0.02$, $C_{n\delta a} = 0.004$, $C_{DS} = 0.40$

of 10 samples, and $t_{\text{pre}} = 0.5\tau$. The steady-state velocities V_v and V_h are 13.5 and 24.8 ft/s, respectively. The parafoil is initially 300 ft upwind of the target with $R = 200$ ft, a $T_{\text{app}}^{\text{des}}$ of 9 s, and a ψ_F of $9\pi/8$. Figure 2 shows the calculated optimal final turn with a tailwind of 10 ft/s ($W_x = 10$ ft/s), and no crosswind. The distance D representing the TIP is 153 ft before the target and the turn time is 25.9 s. Two simulations are shown. The first is the 6-DOF system tracking the desired final turn while the winds remain the same as when the final turn began. In this case, the ideal trajectory is closely tracked and the target is reached. Although the open-loop trajectory planner performs well when the winds are known, in practice, uncertain winds will disturb the ideal solution. The second case shown in Fig. 2 likewise develops the same final turn based on the current winds of $W_x = 10$ ft/s and $W_y = 0$ ft/s, however, after entering the final turn, the wind rotates counterclockwise 0.175 rad and increases in magnitude by 20% such that $W_x = 11.8$ ft/s and $W_y = -2$ ft/s. The increased wind and change in direction pushes the parafoil off the planned trajectory and substantially moves the final approach point. The resulting touchdown is 75 ft away from the target. As expected, although the simplified path planner provides a reasonable strategy for identifying a solution in well-known winds, an effective terminal guidance strategy must compensate for varying winds during the final turn.

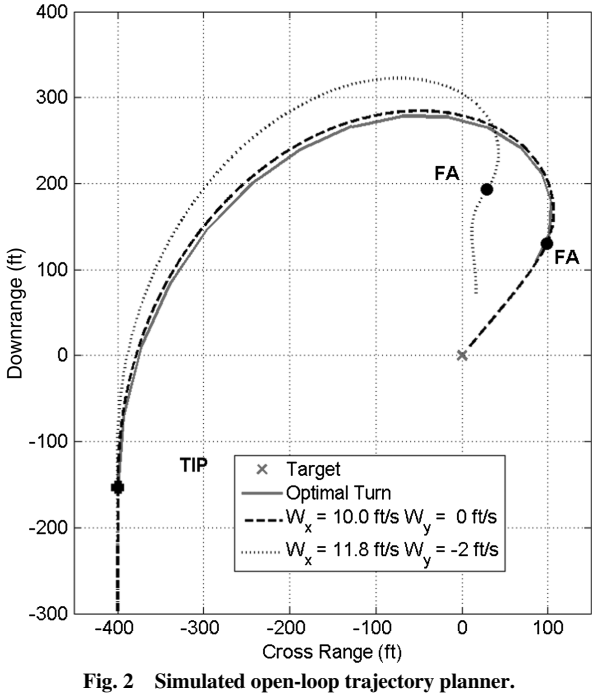

Fig. 2 Simulated open-loop trajectory planner.

III. Robust Terminal Guidance Using General Purpose GPU Computing

Adapting terminal guidance for varying winds has been investigated using a variety of methods [2,3,12,16]. From Fig. 2, it is seen that a possible solution to the increased wind is to decrease the final approach direction and decrease the turn time, resulting in a shorter path to the target. Similarly, if the wind decreases after entering the final turn, the final approach angle could be increased and turn time increased so that a longer path is taken to the target. In either case, the terminal guidance problem must be updated after the final turn has begun based on the current wind knowledge and the parafoil's current state.

A. Adaptive Path Planning

The final portion of terminal guidance first described in Fig. 1 is reconsidered in detail in Fig. 3 where, if the winds and parafoil model are known exactly, the previous open-loop planner would bring the parafoil to the target at touchdown. However, variations in the wind and model will cause the current parafoil position (x_0, y_0, z_0) and heading ψ_0 to vary from the states that satisfy the terminal guidance problem. The problem of finding a new terminal guidance solution

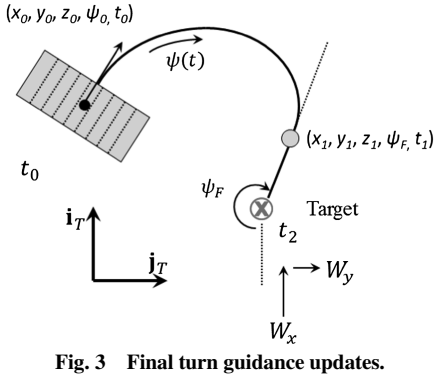

Fig. 3 Final turn guidance updates.

while in the final turn can be summarized as follows: Given a parafoil at (x_0, y_0, z_0), at time t_0, with angle ψ_0, find $\dot{\psi}$ and ψ_F such that the parafoil will land at the target in the presence of winds W_x and W_y.

Because of the problem constraints, there is at most one $\dot{\psi}$ and ψ_F that will satisfy the problem. Therefore, based on the kinematic model (1) and assuming a constant turn rate $\dot{\psi}$ from t_0 to t_1 and a constant ψ_F from t_1 to t_2, a nonlinear optimization problem can be formulated as follows:

Given $x_0, z_0, z_0, \psi_0, t_0$ and estimates of W_x, W_y, find $\dot{\psi}$ and ψ_F that solves

$$\min J = f(\mathbf{s}, t) \tag{10}$$

Subject to

$$\begin{aligned} x_2 &= x_0 + W_x(t_1 - t_0) + \frac{V_h}{\dot{\psi}}(\sin\psi_F - \sin\psi_0) \\ &\quad + (t_2 - t_1)(W_x + V_h\cos\psi_F) \\ y_2 &= y_0 + W_y(t_1 - t_0) + \frac{V_h}{\dot{\psi}}(\cos\psi_F - \cos\psi_0) \\ &\quad + (t_2 - t_1)(W_y + V_h\cos\psi_F) \\ t_2 &= \frac{-z_0}{V_z} + t_0 \\ t_1 &= \frac{\psi_F - \psi_0}{\dot{\psi}} + t_0 \end{aligned} \tag{11}$$

The cost function $f(\mathbf{s}, t)$ may be any desired nonlinear function of time and terminal parafoil states $\mathbf{s}$, such as position, speed, and orientation. For instance, this cost function may penalize strictly impact miss distance, or it may penalize a weighted sum of the miss distance and other statistical measures of the impact states. This two-parameter nonlinear optimization problem must be solved numerically and suffers from the well-known problems of any gradient-based iterative solver, such as convergence, solution speed, and robustness. A fundamentally different solution procedure is presented next based on GPU computing, which provides substantially more flexibility and robustness and augments the final solution based on stochastic information.

B. General Purpose GPU-Based Guidance Solution

As discussed earlier, most precision placement strategies exclusively emphasize touchdown error. Although error is clearly a concern, practical airdrop systems must also include other criteria that may be as important or even more important, such as touchdown velocity for sensitive payloads or geometric constraints for space-limited or complex drop zones. Some limitations of solving the preceding nonlinear optimization problem include the fact that the solution is based on an ideal representation of the problem, and the solution is deterministic. A potential solution may be appropriate based on the known mean winds; however, it could be extremely sensitive to variations in the wind, with a small change resulting in potentially large errors or a large increase in the cost function. As an example, consider a constrained drop zone where, because of a canyon or obstacle, a region short and left of the desired target is forbidden. Using the deterministic solution procedure, the optimal solution may occur close to, but not in, the forbidden region and selected as suitable. However, in the presence of uncertain winds, many of the solutions may result in impact violation. A probabilistic solution would determine which trajectories are sensitive to wind variations and as a result might select a different approach direction that reduces geometric violations at the expense of final touchdown error. Because wind is uncertain, in the case of gusty conditions, it is more important to select a terminal trajectory that is robust to the uncertainties rather than an optimal accurate trajectory assuming perfect knowledge of winds.

One method to rapidly compute system response to stochastic perturbations of a nonlinear plant is to perform efficient Monte Carlo simulations. The rise of GPU computing for general purpose applications has, for the first time, enabled such analyses to be performed as part of a closed-loop real-time control system. GPUs are single-program multiple-data computing devices that differ fundamentally from CPU-type processors in that they typically employ 1–2 orders of magnitude more cores running in parallel.

Originally developed to execute the highly parallel graphics pipeline, GPUs have evolved to become very efficient multithreaded processors with high memory bandwidth. In comparison with CPUs, graphics processors focus less on data caching and flow control, instead emphasizing computational throughput. This design difference explains why GPUs are significantly slower in comparison with CPUs when executing many overhead-intensive tasks (such as running operating system commands), but exhibit far superior performance when executing parallel mathematical computations. For general purpose computing applications that have embarrassingly parallel structure, the run time reductions enabled by GPU computation can reach 1–2 orders of magnitude.

The algorithm proposed here first brings the parafoil to the terminal stage of guidance based on the open-loop trajectory planner described in Sec. II and, upon beginning the final turn, leverages GPU computations to derive impact point statistics resulting from uncertainty in wind estimates. The resulting controller then emphasizes robust guidance as opposed to optimal guidance, assuming perfect knowledge of winds. The proposed robust parafoil terminal guidance algorithm that begins at the start of the final turn includes four elements. The first element is wind estimation and measurement, which occurs during energy management, homing, and terminal guidance stages, producing both a nominal wind and measurement variance throughout terminal flight. Second, generation of several candidate trajectories is completed based on the nominal wind conditions and the nonlinear optimization problem. The third element leverages GPU computations by evaluating the selected candidate trajectories using nominal wind conditions through Monte Carlo simulation, during which their robustness to uncertainty in wind is established. Finally, a robust candidate trajectory is chosen based on a cost function that balances user-defined criteria such as mean impact error e and a function of other statistics $f(\boldsymbol{\sigma})$, where $\boldsymbol{\sigma}$ can include quantities such as the probability of geometric constraint violation, mean impact velocity, variance of impact velocity, etc. A diagram of the four elements of the proposed robust parafoil terminal guidance algorithm is shown in Fig. 4 with a detailed description of each element provided next.

1. *Wind Estimator*

Precision airdrop systems typically integrate wind information and measurements from three types of sources: online estimation, a priori information such as a mission planner or dropsonde, and, more recently, ground-based Laser Detection and Ranging sensors. The wind estimator used in all following simulations assumes that data are synthesized from any combination of the three elements and wind

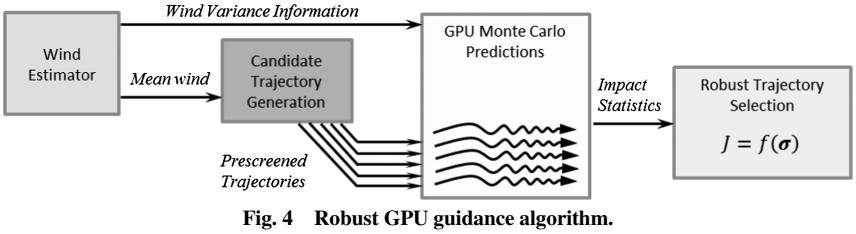

Fig. 4 Robust GPU guidance algorithm.

components are estimated continuously during energy management, homing, and terminal guidance, thus both a nominal wind and measurement variance can be obtained. Note that only horizontal wind components are considered here because their integrated effect along the terminal trajectory tends to be greater than short-time-scale vertical gust perturbations. To represent realistic errors in estimated winds, the actual horizontal wind components were perturbed with Gaussian noise in magnitude and direction at each simulation time step and fed into a 150-sample moving average filter, which produced realistic wind estimates updated at 1 Hz. These wind estimates were fed into the parafoil guidance system for use in candidate trajectory generation and real-time Monte Carlo analysis. For all the cases in this paper, a noise standard deviation of 6.0 ft/s and 0.35 rad was used, respectively, for the magnitude and direction noise added to the simulated winds. Noise added to the wind magnitude was also biased using a randomly generated bias with standard deviation of 0.5 ft/s.

2. *Candidate Trajectory Generation*

This element uses the nominal wind estimate to create a set of candidate trajectories using the nonlinear optimization problem. Rather than using numerical iteration to find the single optimal solution, the solution to M possible turn rates and N final approach directions are found using Eq. (11). The $M \times N$ solutions are prescreened using Eq. (10) to find the best 390 solutions. In this case, the value 390 is selected because optimal GPU execution occurs when Monte Carlo simulations are run in multiples of the number of multiprocessors on the device [23], which, for this particular GPU, is 30. Any number of solutions could be selected, however, the prescreened cases will be sent to the GPU model to evaluate statistics through Monte Carlo simulation for each particular combination of turn rate and approach direction. Therefore, a tradeoff exists between selecting a rich set of possible solutions and the required solution time of the GPU. A powerful feature of the algorithm is the flexibility of the prescreening process. Depending on the user's requirements, the prescreening criterion in Eq. (10) could be simply the impact error, or a weighted impacted velocity could be included, or, in the case of a drop-zone constraint, the minimum error of trajectories that do not violate the constraint may be selected. In any of these examples, the same $M \times N$ solutions are still evaluated, only the prescreening decision needs to be changed.

3. *Massively Parallel GPU Predictions*

The GPU Monte Carlo algorithm is similar to that proposed in [23], in which the GPU kernel is a 6-DOF trajectory propagator. Here, the kernel propagates the 6-DOF parafoil model with the complete MPC guidance system that tracks the candidate trajectory. Blocks of 256 kernels are executed, with one of the 390 prescreened candidate trajectories mapped to each block. Within a block, each of the 256 kernels is assigned a wind value that deviates from the nominal wind estimate based on the estimated wind distribution. As in [23], CUDA-based trajectory codes may be optimized to reduce run time through the use of shared memory caches and intrinsic mathematical functions. As with the number of candidate trajectories, the 256 kernels were selected so as to minimize GPU run time while providing sufficient statistical information. The purpose of the kernels is to provide an approximate statistical representation of each candidate trajectory with respect to horizontal wind variations, and it was determined heuristically that increasing the number of kernels past 256 had limited effect on the statistics, and thus the additional computational time was not warranted. The total number of trajectory predictions at each time step was therefore $390 \times 256 = 99,840$.

4. *Robust Trajectory Selection*

A cost function can be constructed for each of the 390 candidate trajectories based on the statistical information from the GPU Monte Carlo data, which exists for each candidate trajectory. In many cases, this cost function may be constructed as a weighted sum between the mean miss distance and any other statistical measure (although there is no strict requirement that mean miss distance be included). Thus, the minimization cost function is computed according to

$$J = e + f(\boldsymbol{\sigma}) \tag{12}$$

where e is the mean miss distance from the target and $f(\boldsymbol{\sigma})$ is any weighted function based on elements of a statistic vector $\boldsymbol{\sigma}$, which can include the standard deviation of the miss distance, standard deviation of the directional error, number of geometric violations, etc. The candidate trajectory with the minimum cost is considered to achieve the optimum balance between impact accuracy and robustness to wind errors given the chosen criterion. This trajectory is then used during terminal guidance for tracking within the MPC algorithm and can be continually refined during terminal maneuvers by repeating this trajectory planning process. In the simulations to follow, this GPU-based path planning was repeated every 2.5 s during the final turn.

One strength of the proposed algorithm is that the final trajectory decision is based on the statistical robustness demonstrated by analyzing a full dynamic model, including the trajectory tracking control and uncertain winds. This is in contrast to most path planners, which determine an optimal trajectory without regard to sensitivity of the trajectory to stochastic effects. Another strength is the flexibility and simplicity in designing the prescreening and final trajectory selection cost functions, in which any number of goals may be mixed in an optimal fashion. For example, if the impact velocity is to be minimized, the mean impact velocity can be simply added to $f(\boldsymbol{\sigma})$ with a large weight. In contrast, many current precision placement algorithms must achieve this result in an indirect manner, such as artificially constraining the final approach directly into the wind regardless of any penalty on miss distance. Likewise, although implementation of an arbitrary no-impact zone is simple in the proposed algorithm, it is challenging in standard parafoil path planners because the probability of impact within this zone due to wind changes cannot be evaluated in real time without a parallel processor like the GPU. The intuitive augmentation of the cost function on the final guidance makes the proposed algorithm well suited for existed mission planning formats [1,6]. Just as target information and wind profiles are currently loaded, drop-zone constraints, impact velocity requirements, and desired error statistics that may vary from system to system can easily be updated before launch of the parafoil.

IV. Results

Demonstration of the proposed algorithm will first be shown through the comparison of two representative implementations of the proposed GPU-based algorithm, with a third case employing the nonadaptive open-loop trajectory planner of Sec. II. All three simulations use the 6-DOF model described previously with x_0 and y_0 of -400 ft and an altitude of 508 ft. The desired turn radius R is 200 ft and the desired final approach direction is set to π rad. The wind is initially 15 ft/s in a direction rotated 0.05 rad clockwise from the x axis. At 14.6 s (the middle of the turn), the wind decreases in magnitude to 7.5 ft/s and rotates to 0.85 rad clockwise from the x axis. Wind estimates during terminal guidance are updated at 1 Hz. It is important to note that, using 256 kernels and 390 candidate trajectories (99,840 total simulations), it takes less than 2 s to complete all calculations with an NVIDIA Tesla C2050-series GPU. Compared with a single CPU, the required run time is reduced by 1–2 orders of magnitude. The adaptive GPU algorithm is therefore updated every 2.5 s after the final turn commences, demonstrating it can be used in real time. Candidate trajectories are found by selecting both M and N to be 100 and using equidistant values of possible turn rates spanning $\pm\pi/4$ rad/s, which is the maximum desired turn rate for this particular parafoil, so that the bank and sideslip angles remain small. The candidate ψ_F is also chosen from a set of equally spaced angles over a range of $7\pi/8 - 5\pi/4$ during the initial turn, then increased to a range of $\pi/2 - 5\pi/4$ during the latter stages of the turn. Figure 5 shows three trajectories, and the wind estimates for all cases are shown in Fig. 6. The baseline uses the open-loop planner, which

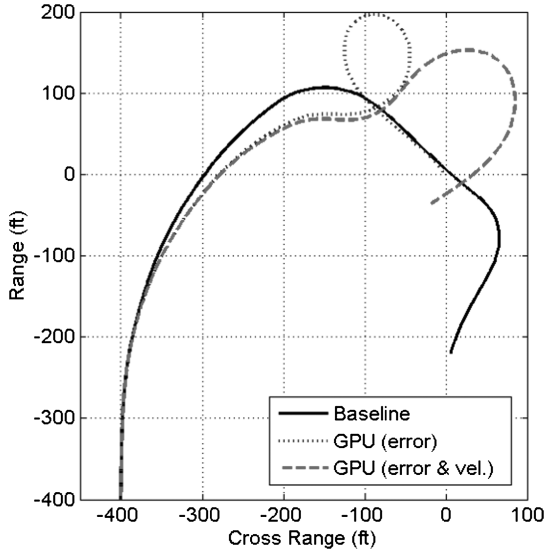

Fig. 5 Terminal guidance trajectories.

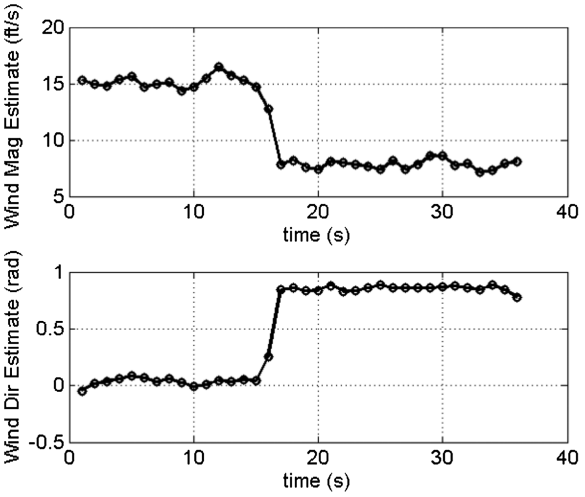

Fig. 6 Estimated wind magnitude and direction.

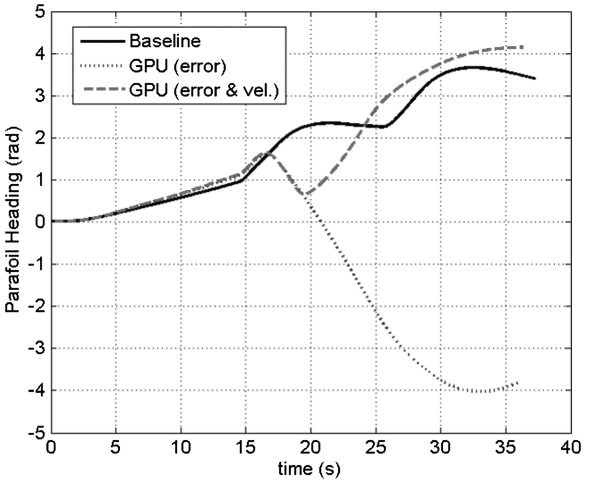

Fig. 7 Parafoil heading angle history.

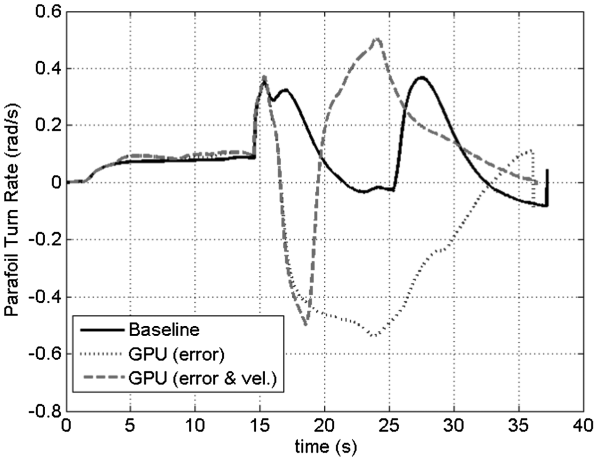

Fig. 8 Parafoil turn rate history.

does not update the trajectory even after detecting the change in wind. The two GPU cases are designated by the labels "error" and "error & vel." In the error case, the cost function is given by

$$J = e \tag{13}$$

where e is the mean impact error from each Monte Carlo case, and prescreening is performed by choosing the 390 cases with the lowest impact error predicted from the first two equations in Eq. (11). In the error and velocity case, the cost function is given by

$$J = e + 30\bar{v} \tag{14}$$

where $\bar{v}$ is the mean impact horizontal velocity from each Monte Carlo case, and prescreening is performed by choosing the 390 cases with the lowest predicted impact velocity.

Trajectories of the three cases are shown in Fig. 5, where the change in wind occurs at approximately the middle of the final turn. The open-loop trajectory planner determines an initial turn rate of 0.12 rad/s at the beginning of the turn. The baseline case responds predictably by overshooting the target by 220 ft when the wind magnitude decreases. Heading and turn rate histories are shown in Figs. 7 and 8 where the abrupt changes to the baseline case occur when the wind gust begins and at the transition from final turn to final approach. The change in wind acts to increase the turn rate from the predicted constant value of 0.12 rad/s. Near impact, the final heading angle approaches the commanded approach angle π as expected.

Implementation of the robust GPU algorithm results in two significantly different trajectories when penalizing error only, compared with penalizing both error and impact velocity as seen in Fig. 5. Time histories of the robust GPU solution for the desired turn rate and approach angle are provided in Fig. 9. Comparison of the

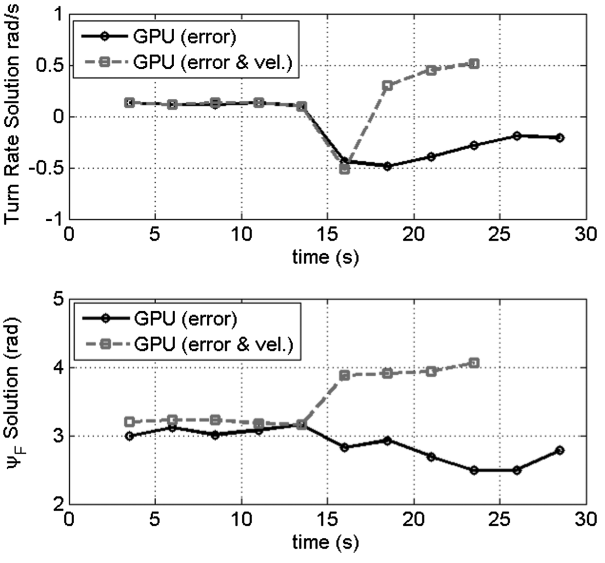

Fig. 9 Robust parafoil terminal guidance.

results for the first 15 s of Fig. 7 shows all three cases appear to be similar. Once the wind gust is detected, however, both adaptive cases initially respond by turning away from the target at −0.5 rad/s to delay the approach. Afterward, the two cases diverge. The GPU case penalizing only error continues a counterclockwise turn until reaching the target almost perpendicular to the wind with only 3 ft of error. In contrast, the GPU penalizing both error and impact velocity chooses to turn back into the wind to land with a final approach of 4.0 rad and a final error of 40 ft. Although the velocity penalty increases the final impact error, the horizontal impact velocity is reduced 25% from 22.9 to 17.3 ft/s. A remarkable feature displayed in Fig. 6 is that no substantial changes to the algorithm are required for either case and yet, based on statistical information of simulations from the decision point to impact, the robust GPU algorithm selects fundamentally different trajectory solutions. A phenomenon of the GPU algorithm seen in Figs. 5 and 7 is that the initial turn rate is always chosen slightly larger than the open-loop case and the turn is made earlier. This reflects the probabilistic results that, with some uncertainty in winds, it is preferred on average to go toward the target slightly early in the event that the wind increases and the ability to reach the target is diminished. If the wind does not increase, it always possible to turn back near the end of the trajectory, thus still achieving small error. Another interesting feature of the GPU algorithm in the case penalizing error only is that the parafoil executes an optimal 2π rad turn during the final approach to expend excess energy, as shown in Fig. 5. There are no open-loop commands within the control system to execute such a turn; rather, such a 2π rad turn turns out to be the optimal solution to the minimization problem posed in Eqs. (10) and (11) and is therefore performed.

Figures 10 and 11 show a dispersion comparison based on 200 simulations of the guidance, not using the GPU and the full GPU-based guidance solution for both impact error only and combined impact error and velocity penalties [i.e., the cost functions and prescreening procedures described in Eqs. (13) and (14), respectively]. Dispersion simulations were generated using a wind gust that can occur in any direction with uniform probability and a standard deviation of 12 ft/s from the nominal wind for that particular case. For comparison, with the parafoil airspeed and defined wind gust, the theoretical minimum average touchdown velocity for these cases is about 14.5 ft/s when the parafoil lands directly into the wind every time. Removal of wind updates and the GPU from the proposed guidance algorithm results in an algorithm similar to existing methods as proposed in [16]. Figure 10 shows the dispersion without the GPU or wind updates, which has a 50% target-centered circular error probable (CEP) of 167 ft and a median

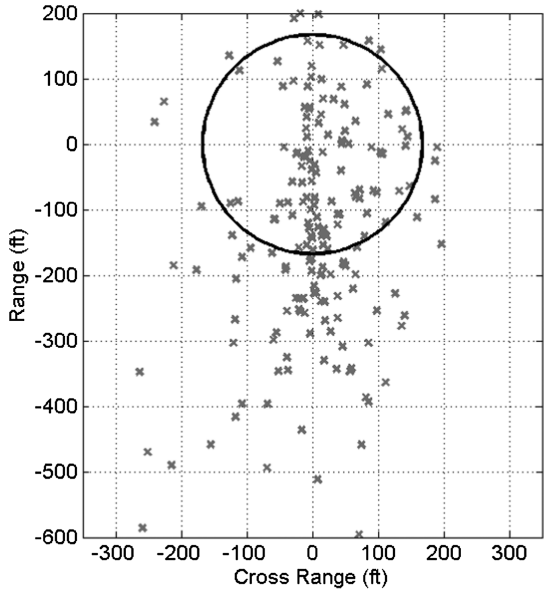

Fig. 10 Dispersion using no GPU or wind updates (X), CEP = 167 ft.

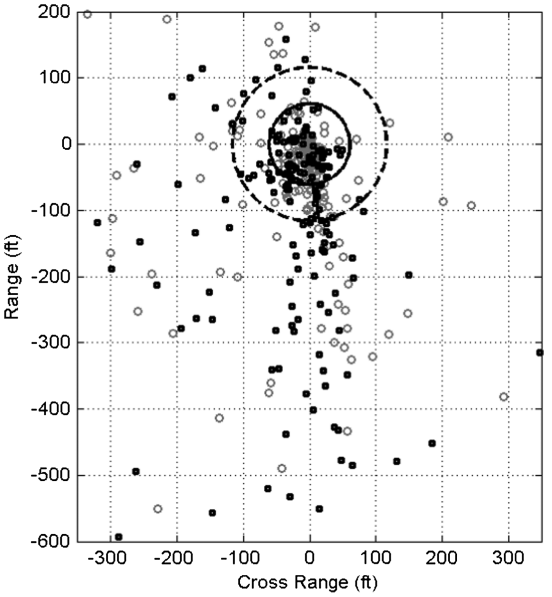

Fig. 11 Dispersion comparison with GPU penalizing error only (O) and both error and velocity (■); CEP = 61 and 116 ft, respectively.

impact velocity of 24.0 ft/s (9.5 ft/s above the theoretical minimum).

Inspection of the dispersion patterns in Fig. 11 for the GPU-based guidance solution, comparing impact error only and combined impact error and velocity penalties, demonstrates that both cases have similar patterns. However, when penalizing only miss distance, the median impact velocity is 20.9 ft/s (6.4 ft/s above the theoretical minimum), whereas in the case when both miss distance and horizontal impact velocity are penalized, the median impact velocity is 17.5 ft/s (3.0 ft/s above the theoretical minimum). Decreasing the horizontal impact velocity comes at the expense of an increase in the CEP, which is 61 ft for the error-only case and 116 ft for the case in which impact velocity is also penalized. In both cases, the adaptive GPU-based guidance outperforms the conventional algorithm in accuracy and impact velocity.

A final demonstration of the algorithm's robustness and flexibility is the addition of a geometric constraint region. A region spanning from −100 to 0 ft cross range and from 25 to 200 ft downrange is defined as a penalty region. For this case, the candidate ψ_F is chosen equidistantly over a full range of 0–2π, allowing the parafoil to approach from any direction. A baseline case uses only impact error in both the prescreening and final cost function, and thus makes no attempt to avoid the penalty region. The geometrically constrained case uses a cost function given by

$$J = e + 1000 p_v \tag{15}$$

where p_v represents the probability that the particular choice of control parameters $\dot{\psi}$ and ψ_F will lead to impact within the penalty region. This probability is approximated by the percentage of cases in the Monte Carlo simulation that impact within the penalty region. Prescreening for this geometric constrained case is accomplished by first eliminating any cases predicted by the kinematic model in Eq. (10) to impact within the penalty region, and then choosing the 390 most accurate.

Figure 12 shows a dispersion comparison based on 200 simulations, generated using a wind gust with a normal distribution that has a standard deviation in magnitude of 10 ft/s from the initial wind for the particular case, and in direction with a standard deviation of 0.7 rad. When the geometric constraint region is not penalized, 39 of 200 impacts (20%) land within the penalty region. In contrast, in the geometric constrained case only 4 of 200 impacts (2%) land within the region. The 50% target-centered CEP is only slightly increased by avoiding the region and changes from 44 to 56 ft.

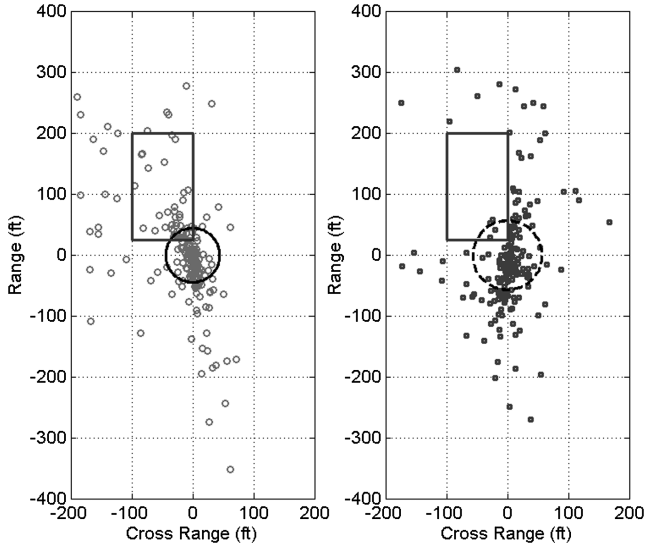

Fig. 12 Dispersion comparison between penalizing error only (O) and both error and a geometric constraint (■).

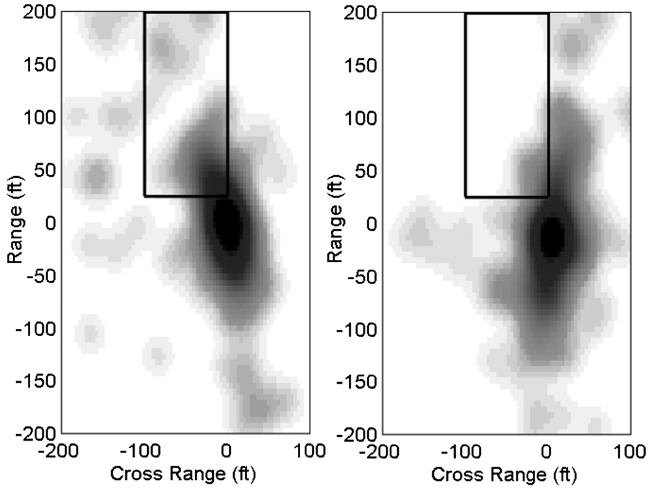

Fig. 13 Probability density map (left, penalizing error; right, penalizing error and geometric constraint).

A probability density map of the two dispersions, shown in Fig. 13, highlights a strength of the algorithm. From this map, it can be seen that the case penalizing error only has a distribution pattern that has a longitudinal axis of about 3 rad, which represents a nearly upwind landing but a small tendency to use a shorter approach to account for the possibility of increasing winds, as described earlier. A conventional approach to avoid the region would be to artificially move the target significantly away from the true target and potentially rotating the landing pattern by as much as 0.96 rad and land with a crosswind, both solutions increasing impact error. In contrast, the robust GPU-based guidance solution, as seen in Fig. 13, shapes the probability density map by rotating the longitudinal axis by only 0.26 rad and moving the mean impact 16 ft to the left, which accounts for the small increase in target-centered CEP. In addition, the misses are now forced to be to the low and right side of the geometric penalty region.

V. Conclusions

A novel parafoil terminal guidance algorithm is presented, designed specifically to improve guidance robustness and enable new control capabilities. Although more computationally efficient feedback trajectory planning laws could be used to penalize error only, complex capabilities such as combined error minimization, velocity minimization, and impact constraints benefit from graphics processing unit (GPU)-based guidance. The algorithm is unique in that it solves online a general nonlinear optimization problem in which any number of statistical parameters can be simultaneously minimized. Furthermore, the algorithm relies on real-time Monte Carlo simulation of a six degree-of-freedom parafoil model under wind disturbances to provide statistical parameters to the general cost function, such as mean impact velocity and the probability of landing within penalty regions. Generation of these impact statistics is enabled by the use of onboard graphics processing units, which are a massively parallel processing architecture well suited to Monte Carlo computation. Results show that, although open-loop trajectory planners may exhibit reasonable performance when winds are constant and known precisely, a change in winds during the final turn may lead to significant error and thus requires adaptation of the prescribed trajectory. However, using the GPU-based control scheme, impact statistics can be continually evaluated through Monte Carlo simulations that account for wind disturbances, and impact error is substantially reduced. Additional examples show that, through simple manipulation of the controller cost function, various velocity or drop-zone geometric constraints may be enforced that are otherwise difficult to implement in standard parafoil guidance schemes.

References

[1] Benney, R., McGrath, J., McHugh, J., Melono, A., Noetscher, G., Tavan, S., and Patel, S., "DoD JPADS Programs Overview and NATO Activities," *19th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA Paper 2009-2952, 2007.

[2] Bergeron, K., Fejzic, A., and Tavan, S., "Acculglide 100: Precision Airdrop Guidance and Control via Glide Slope Control," *21st AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA Paper 2011-2530, 2011.

[3] Ward, M., "Adaptive Glide Slope Control for Parafoil and Payload Aircraft," Ph.D. Dissertation, Georgia Inst. of Technology, Atlanta, 2012.

[4] Benney, R., Meloni, A., Cronk, A., and Tiaden, R., "Precision Airdrop Technology Conference and Demonstration," *20th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA Paper 2009-2927, 2009.

[5] Tavan, S., "Status and Context of High Altitude Precision Aerial Delivery Systems," AIAA Paper 2006-6793, 2006.

[6] Wright, R., Benney, R., and McHugh, J., "Precision Airdrop System," AIAA Paper 2005-1644, 2005.

[7] Calise, A., and Preston, D., "Design of a Stability Augmentation System for Airdrop of Autonomous Guided Parafoils," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2006-6776, 2006.

[8] Calise, A., and Preston, D., "Approximate Correction of Guidance Commands for Winds," *20th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA Paper 2009-2997, 2009.

[9] Gimadieva, T. Z., "Optimal Control of a Guiding Gliding Parachute System," *Journal of Mathematical Sciences*, Vol. 103, No. 1, 2001, pp. 54–60.

[10] 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," *19th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA Paper 2007-2514, 2007.

[11] Slegers, N., and Costello, M., "Model Predictive Control of a Parafoil and Payload System," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 816–821. doi:10.2514/1.12251

[12] Slegers, N., Beyer, E., and Costello, M., "Use of Dynamic Incidence Angle for Glide Slope Control of Autonomous Parafoils," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 585–596. doi:10.2514/6.2007-2526

[13] 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

[14] Carter, D., Singh, L., Wholey, L., Rasmussen, S., Barrows, T., George, S., McConley, T., Gibson, T., Tavan, S., and Bagdonovich, B., "Band-Limited Guidance and Control of Large Parafoils," *20th Aerodynamic Decelerator Systems Technology Conference and Seminar*, AIAA Paper 2009-2981, 2009.

[15] Rademacher, B., Lu, P., Strahan, A., and Cerimele, C., "In-flight Trajectory Planning and Guidance for Autonomous Parafoils," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 6, 2009, pp. 1697–1712. doi:10.2514/1.44862

[16] Slegers, N., and Yakimenko, O., "Terminal Guidance of Autonomous Parafoils in High Wind-to-Airspeed Ratios," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 225, No. 3, 2011, pp. 336–346. doi:10.1243/09544100JAERO749

[17] Liu, W., Schmidt, B., Voss, G., and Muller-Wittig, W., "Molecular Dynamics Simulations on Commodity GPUs with CUDA," *High Performance Computing*, Lecture Notes in Computer Science, Vol. 4873, 2007, pp. 185–196.

[18] Friedrichs, M., Eastman, P., Vaidyanathan, V., Houston, M., Legrand, S., Beberg, A., Ensign, D., Bruns, C., and Pande, V., "Accelerating Molecular Dynamic Simulation on Graphics Processing Units," *Journal of Computational Chemistry*, Vol. 30, No. 6, 2009, pp. 864–872. doi:10.1002/jcc.v30:6

[19] Jacobsen, D., Thibault, J., and Senocak, I., "An MPI-CUDA Implementation for Massively Parallel Incompressible Flow Computations on Multi-GPU Clusters," *48th AIAA Aerospace Sciences Meeting*, AIAA Paper 2010-522, 2010.

[20] Stock, M., and Gharakhani, A., "Toward Efficient GPU-Accelerated N-Body Simulations," *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2008-608, 2008.

[21] Grossauer, H., and Thoman, P., "GPU-Based Multigrid: Real-Time Performance in High Resolution Nonlinear Image Processing," *Lecture Notes in Computer Science*, Vol. 5008, Computer Vision Systems, Springer, Berlin, Heidelberg, 2008, pp. 141–150.

[22] Phillips, E., Davis, R., and Owens, J., "Unsteady Turbulent Simulations on a Cluster of Graphics Processors," *40th Fluid Dynamics Conference and Exhibit*, AIAA Paper 2010-5036, 2010.

[23] Ilg, M., Rogers, J., and Costello, M., "Projectile Monte Carlo Trajectory Analysis Using a Graphics Processing Unit," *2011 AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2011-6266, 2011.

[24] "OMAP 5 Platform—OMAP5430," Texas Instruments, Dallas, TX, http://www.ti.com/general/docs/wtbu/wtbuproductcontent.tsp?templateId=6123&navigationId=12863&contentId=103102 [retrieved 18 July 2012].

[25] Gorman, C., and Slegers, N., "Model Predictive Control of Continuous Nonlinear Systems Using Series Approximations," *International Review of Automatic Control*, Vol. 2, No. 3, 2009, pp. 249–257.

[26] Slegers, N., Kyle, J., and Costello, M., "A Nonlinear Model Predictive Control Technique for Unmanned for Air Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 5, 2006, pp. 1179–1188. doi:10.2514/1.21531

[27] Gorman, C., and Slegers, N., "Comparison and Analysis of Multi-Body Parafoil Models with Varying Degrees of Freedom," *21st AIAA Aerodynamic Decelerator Systems Technology Conference*, AIAA Paper 2011-2615, 2011.