JOURNAL OF AIR TRANSPORTATION
Vol. 27, No. 3, July 2019

Optimal Path Planning for Unmanned Aircraft Target Observation Through Constrained Urban Environments

Michael D. Zollars* and Richard G. Cobb†
Air Force Institute of Technology, Wright–Patterson Air Force Base, Ohio 45433
and
David J. Grymin‡
Controls Science System Center of Excellence, Wright–Patterson Air Force Base, Ohio 45433

DOI: 10.2514/1.D0141

The work herein determines the optimal flight path for a small unmanned aircraft system through a constrained urban environment. The implementation of unavoidable keep-out regions is evaluated by first solving the optimal flight path through a constrained simplex corridor and, second, minimizing incursions to keep-out regions given the same constraint field and ending on a defined orbit around a target of interest. Direct orthogonal collocation methods are combined with fast geometric path planning techniques where a triangulated mesh is used to produce a hybrid control routine, resulting in optimal flight paths through a defined triangulated corridor. Physical constraints are eliminated from the nonlinear program search space, while keep-out regions are modeled within the objective function of the optimal control problem and avoided according to a weighted distribution of the objective components. A scenario is presented for a small unmanned aircraft system to advance at constant altitude and constant speed through city building constraints while minimizing time in unavoidable keep-out regions. The path terminates outside the triangulated corridor on an orbit, encircling the target location. Results illustrate an optimal path solution through 37 polygonal constraints and 2 nonlinear, unavoidable keep-out regions with computational times on the order of 90 s.

Nomenclature

a	=	lateral superellipse axis, ft
b	=	longitudinal superellipse axis, ft
C	=	constrained edges
J	=	cost functional
Kx	=	keep-out region x-coordinate
Ky	=	keep-out region y-coordinate
P	=	total number of phases
R	=	orbit radius
s	=	stiffness parameter
t	=	mission time
$\boldsymbol{u}$	=	control vector
$\boldsymbol{X}$	=	state vector barycentric reference
$\tilde{X}$	=	state vector Cartesian reference
x	=	position along the x-axis, ft
$\dot{x}$	=	velocity in the x-axis, ft/s
$\tilde{x}$	=	final SUAS x-axis position
y	=	position along the y-axis, ft
$\dot{y}$	=	velocity in the y-axis, ft/s
$\tilde{y}$	=	final SUAS y-axis position
α	=	barycentric weights
β	=	cost function weight parameter
λ	=	constraint line segment slope
μ	=	orbit direction
ζ	=	constraint minimum line segment
v	=	SUAS airspeed, ft/s
ρ	=	time-bound scaling factor
θ	=	SUAS heading angle, rad
$\dot{\theta}$	=	SUAS heading angle rate, rad/s

Presented as Paper 2018-2259 at the 2018 AIAA Information Systems-AIAA Infotech @ Aerospace, Kissimmee, FL, 8–12 January 2018; received 3 August 2018; revision received 18 March 2019; accepted for publication 19 March 2019; published online 22 April 2019.

*Ph.D., Department of Aeronautics and Astronautics, 2950 Hobson Way, Lt. Colonel, USAF. Member AIAA.
†Professor, Department of Aeronautics and Astronautics, 2950 Hobson Way, USAF. Associate Fellow AIAA.
‡Controls Research Engineer, Air Vehicles Directorate, AFRL.

I. Introduction

SMALL unmanned aircraft systems (SUAS) have played a major role in mission contributions across the Department of Defense. Their impact continues to be recognized as their role transitions from remote-piloted operations and waypoint following to autonomous navigation. Manned-unmanned teaming has become a research priority with goals set to demonstrate effective human-machine interactions while increasing the trust of autonomous missions [1]. These SUAS, performing autonomously, have the potential to use sensor packages in close proximity to areas of interest and provide reliable data to manned aircraft located at a safe off-set distance.

Within the construct of autonomous flight path planning, direct orthogonal collocation methods have significantly increased computation efficiencies when an adequate initial guess is used to seed the nonlinear program (NLP) solver and parameter bounds are accurately represented [2–5]. Leveraging fast geometric path planning algorithms, the optimal control problem can be formulated within a triangulated mesh using a barycentric coordinate system, thus standardizing parameter bounds within each simplex. The triangulated mesh eliminates nonlinear path constraints from the search space and a Dubins path solution through a triangulated Connected Simplex Corridor (CSC) is generated. The result is a CSC that has effectively removed nonlinear constraints from the NLP search space and standardizes the bounds of each simplex, regardless of the size of the global domain, while producing an adequate solution for seeding the NLP.

Previous work has shown the implementation of the CSC method through highly constrained environments but has only considered constraints as hard building constraints that are removed from the search space. This body of work focuses on the development of a feasible flight path through a constrained environment quickly and efficiently with a hybrid method combining optimal control direct orthogonal collocation methods with fast geometric path planning techniques while minimizing incursions to unavoidable keep-out regions within the search space. These keep-out region constraints must be accounted for within the CSC and may overlap simplex boundaries. The vehicle is required to end on a path coincident to a

final desired orbit with a heading vector perpendicular to the desired target orbit. Event constraints are implemented to enforce a terminal condition where the terminal phase of the scenario requires transition from barycentric coordinates to the feasible regions in global Cartesian coordinates to allow for an optimal trajectory to the final orbit.

The scope of the work herein is limited to the constrained trajectory optimization motivated by increasing computation speeds, accuracy, and convergence rates of the optimal control software to begin to bridge the gap for implementing optimal solutions on real-time systems. The solutions is limited to two-dimensional flight as fast geometric path planners operating in three dimensions have not reached a maturity level adequate for the methodology explored in this work. Further, all simulations are conducted in the open loop focused on generating the initial trajectory from the start point to the finish. Modeling of the closed-loop system to account for mismodeled dynamics, sensor noise, and exogenous inputs, to include wind disturbances, is outside the current scope of this research.

II. Background

A. Direct Orthogonal Collocation

Many methods have been developed to transcribe an infinite-dimensional optimal control problem into a finite optimal control problem, or NLP. A few of these include shooting methods [6], state and control parameterization methods [7], and direct orthogonal collocation methods [2,8]. Direct orthogonal collocation methods are accomplished by first approximating the continuous functions of the optimal control problem with a finite-dimensional Lagrange polynomial basis where the state vector is approximated at a set of collocated points described as

$$x(\tau) \approx \tilde{x}_N(\tau) = \sum_{i=1}^{n+1} x_i L_i(\tau) \tag{1}$$

where $\tilde{x}_N$ is the N point approximation of $x(\tau)$, x_i represents the weighting function, and $L_i(\tau)$ is the Lagrange polynomial basis

$$L_i(\tau) = \prod_{\substack{j=1 \\ j\neq i}}^{n+1} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{2}$$

Here, τ represents an affine transformation of the time t on the interval from $(-1, 1)$ and τ_j represents the n Legendre-Gauss-Radau points, $(\tau_1, \tau_2, \ldots, \tau_n)$, where $\tau_1 = -1$ and $\tau_{n+1} = 1$ [9–11]. Differentiation or integration of the state and control is then calculated through Gaussian quadrature. This method is termed *global* as each collocation point is solved simultaneously. However, in order to fully take advantage of the efficiencies of Lagrange polynomials and Gaussian quadrature, a quality guess of the states, control, and time must be provided to allow for efficient convergence of the optimal control problem.

Direct orthogonal collocation methods have shown to be successful in a variety of applications to include aircraft collision avoidance [2], loyal wingman [12], hypersonics [4], missile avoidance [5], Air Force range flight safety [13], and many more in the aerospace field. In each of these examples, computation speeds and convergence issues have prevented the algorithms from executing onboard the vehicle in real-time operations. These issues are attributed to the difficulties in providing an adequate initial guess to seed the NLP, modeling polygonal-shaped constraints, invalid solutions due to collocation points passing through constraint edges, and the difficulty of modeling problem-specific bounds on the states, control, and time. Humphreys began to address these problems by using a particle swarm optimization algorithm to seed the NLP and publishes computation times for minimizing incursions to keep out regions with a simplistic scenario [14]. His results show the benefit of using a heuristic to seed the NLP as well as the computation times incurred from a single phase when minimizing incursions to a region.

B. Constraint Models

The complexity of the optimal control problem can grow exponentially when constraint models are incorporated. These models can range from simplistic shapes representing circular or elliptical regions, to superquadrics, or even polygonal shapes modeled with Boolean functions [15]. Each of these must be modeled as a path constraint in the optimal control problem, reducing the sparsity of the Jacobian matrix and increasing the computational requirements. Further, these path constraints must be smooth differentiable functions in order to quickly acquire the Jacobian and Hessian. This can be problematic when designing algorithms to handle multiple constraints in a computationally efficient fashion.

For problems where the constraints must be included in the optimal control problem, previous work [2,14] has shown the benefits of combining superellipse functions with sigmoid functions. The superellipse centered at (0, 0) is defined by

$$F(x, y) = \left(\frac{x}{a}\right)^N + \left(\frac{y}{b}\right)^M \tag{3}$$

where any point on the SUAS trajectory is outside of the constraint when $F > 1$. Variables a and b represent the semi-major and semi-minor axes of the superellipse, and $N \geq 2$ and $M \geq 2$ are even numbers representing the curvature of the shape. These shapes are beneficial in the flexibility to model general constraint structures; however, as N and M are increased, the gradient of the function becomes excessively large at the corners and the function itself can grow without bound, resulting in a poorly scaled function. To minimize the impacts this can have on an NLP solver, the function is incorporated into a modified inside-outside function through a sigmoid function [14]. This allows for a bounded, continuous, and differentiable function.

$$\phi(F) = (1 + e^{(s(F(x,y)-1))})^{-1} \tag{4}$$

Here, s represents the stiffness parameter of the curve and F as defined in Eq. (3). Figure 1 illustrates a scaled representation of a straight-line SUAS trajectory through a keep-out region centered at (1, 1), annotated with asterisks. The gray curves depict the stiffness of the function as s is varied from 0.1 to 10.

Incorporating this function as a path constraint, any functional value greater than zero represents a position close to or inside the keep-out region dependent on the stiffness parameter. The normalization of this function value can be handled through the distributed

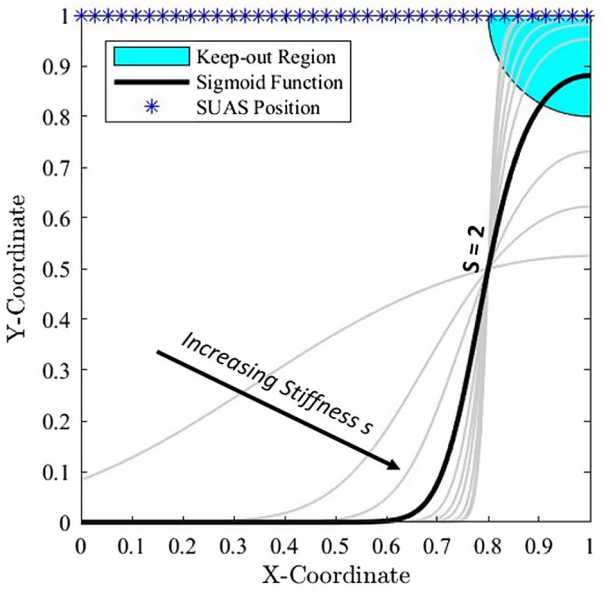

Fig. 1 Sigmoid curves for varying stiffness values.

weight values when included in a cost function. This work implements a stiffness value of $s = 2$, illustrated by the solid black line in Fig. 1. This allows for some incursion to the keep-out region while maintaining an adequate distance from the center point of the threat. The value of this parameter is subjective and should be dependent on the severity of the threat in the region.

C. Path Planning Search Algorithms

Myers et al. executed the Dijkstra search algorithm to find minimum paths for SUAS through polygon obstacles [16]. Their research focused on the two-dimensional search with basic vehicle dynamics in a node-to-node search, resulting in a Dubins path model. They cite difficulties guaranteeing feasible paths and minimizing computational time, whereas the Dijkstra algorithm only allows for finite increments in the vehicle control for heading with a constant speed. Further, the Dijkstra search algorithm implemented in Myers' research does not account for flight dynamics, and therefore a scenario could unfold where the CSC through a constraint field is infeasible due to the rate limits on the vehicle state parameters.

A hybrid solution for optimal control was developed by Aoude using an rapidly exploring random tree algorithm to develop an initial guess for an optimal control solution by pseudospectral methods [17]. The goal of his research was to develop real-time path planning for multiple spacecraft reconfiguration maneuvers with various path constraints. His results show the benefit of using an RRT algorithm as times were drastically reduced, but did not reach a level required for onboard operations and a consistency in solution convergence was not achieved [18].

When considering the constrained constant-altitude SUAS flight trajectory problem, parallels can be made to path planning techniques used in fast geometric path planners. In computer animation, virtual worlds are populated with autonomous agents that are required to traverse through simulated environments while avoiding all obstacles. The final result must be computationally efficient while taking into account path length, time, and energy expended to produce a realistic simulation operated in a real-time environment [19].

An extensive review of path planning though virtual environments with clearances is given in [20]. Algorithms to determine shortest paths while providing a minimum clearance to all constraints are provided in [21]. These algorithms focus on providing reliable, computationally efficient path solutions and provide a framework for dynamically removing or adding constraint regions. The work herein implements these algorithms in the 2010 version of the Triplanner toolkit.[§] A more extensive review of the algorithm can be found in [20,22].

D. Optimal Solutions Through a Simplex Mesh

Previous work has shown the impact constraint functions can have on computation time and accuracy of an optimal control solver as well as the difficulties in efficiently modeling a large number of constraints within the NLP. Initial research has demonstrated the effectiveness of eliminating constraints from the optimal control problem and providing a feasible initial guess posed through a simplex mesh and compared the results to traditional constraint functions [15]. This work was extended in a more realistic two-dimensional environment to illustrate the effectiveness and efficiencies gained by applying a CSC structure through a constrained urban environment [23]. Narrow corridors formed between building structures often become restricted when the vehicle is required to maintain a single speed throughout the mission profile. Additional work explored these tight path corridors by including speed control on the SUAS. Reducing the minimum turn radius arcs to correspond to the vehicle's minimum speed in the geometric path planner increased the potential search space and allowed for a search corridor through tightly constrained spaces [24]. Finally, path planning for multiple waypoints demonstrated the methodology of linking geometric path planning solutions and building appropriate data structures to solve the optimal control problem designed to reach multiple defined locations in the urban environment [25]. The work herein expands on the previous research detailed above by including constraint functions in the NLP to allow minimal incursions to unavoidable keep-out regions and applying terminal constraint conditions to the flight path. This work expands the mission profile to include scenarios with surface-to-air ground threats, restricted flight space, and terminating the flight path on a defined orbit around a target of interest. Each of these constraints is applied herein within the simplex mesh construct.

[§]Data available online at http://graphics.ucmerced.edu/software/tripath/.

III. Methodology

Methodologies to eliminate path constraints from the search domain by defining the problem within the construct of a simplex mesh are evaluated herein. This work extends the basic constrained path planning problem to consider unavoidable constraints designed as keep-out regions where the aircraft must minimize incursions to the constrained airspace.

Figure 2 describes the components and flow of data for implementation of this methodology. The algorithm is initiated by defining the mission parameters to include the vehicle-specific bounds on the state and control variables, the initial and final points of the flight mission, the vertex points of the control space, and the vertex points of each constraint that must be avoided. These user inputs are given to the fast geometric path planner where the path is discretized into a simplex mesh, a CSC is defined, and an initial Dubins path is determined. These data are then conditioned and formatted into a matrix containing the initial solution for the time, states, and control. This matrix is partitioned into each simplex of the CSC and a connectivity matrix defines the proper order of the channel. These data are provided to the NLP solver, where a transformation to the barycentric coordinate system is performed to allow for the implementation of a phased approach to the optimal control problem.

A. Initial Path Solution

Fast geometric path planning algorithms, such as the Triplanner toolkit, have been designed to formulate path solutions in constrained, two-dimensional environments with computational times on the order of milliseconds [20]. The input for the search algorithm requires the vertex of each constraint in the environment and the initial and final position. A constrained delaunay triangulation is performed on the space and is refined to maximize potential search corridors within the mesh. The feasible space is evaluated by including circular constant radius arcs on each vertex point to account for the minimum turning radius of the SUAS defined by the constant speed of the vehicle. Their inclusion allows for feasible flight paths to be determined within the search space. In regions where the circles

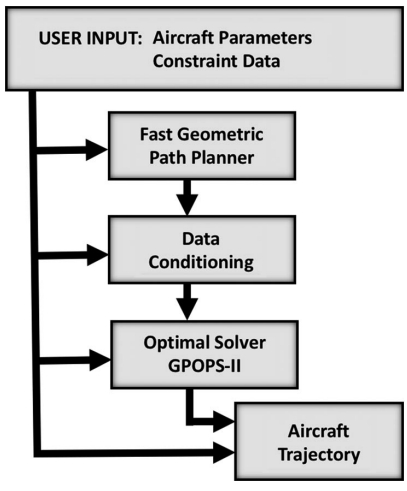

Fig. 2 Simplex methodology.

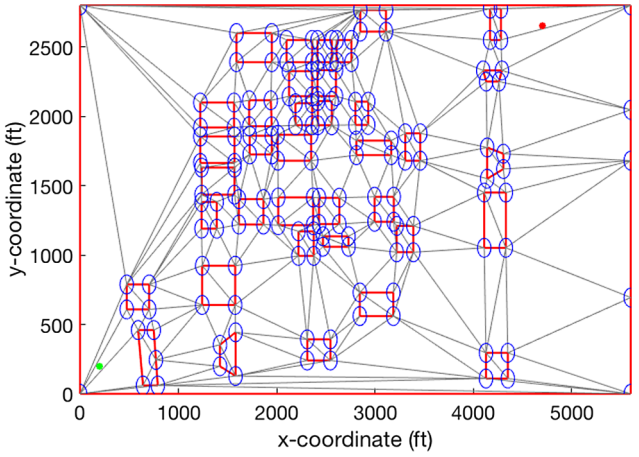

Fig. 3 Triplanner constraint map.

intersect, no guarantee is given for a feasible flight path, and therefore that region is not considered for the CSC.

A portion of downtown Chicago, USA, is depicted in Fig. 3 and shows polygon constraints defining buildings with heights greater than 550 ft above ground level (AGL) and circular arcs attached to each vertex point. The vehicle is required to transit from the bottom left corner of the map, through the constrained field, ending at the target in the top right corner, each defined by the respective asterisks.

Finally, an A* search algorithm [26] is executed to determine a set of simplexes that form a corridor from the starting location to the final location. This path defines a polygonal-shaped corridor that is free of constraints. The autonomous agents' path is determined with a funnel algorithm [27], which excludes the interior of the constant radius circles, resulting in a feasible flight path through a defined search corridor, or CSC. The path resembles a Dubins path solution as a series of straight-line segments connecting the tangent points of the appropriate circles. An example of this CSC and resulting Dubins path is shown in Fig. 4.

Fast geometric path planners provide a fast, feasible solution to the path planning problem; however, limitations to the algorithms do exist. First, geometric path planners do not account for limitations on system parameters and control. Second, unavoidable keep-out regions cannot be handled with geometric path planning algorithm as the triangulated mesh eliminates all constraints from the search space. Finally, geometric path planners provide a point-to-point solution and do not satisfy the terminal requirement to optimally end the path on a nondeterministic point of the desired orbit around the target. These issues must be overcome before an optimal solution can be determined.

To generate the initial path from the geometric path planner, the Dubins path solution is used for the position states with the aircraft dynamics accounted for by a radial offset distance from each constrained edge equal to the SUAS turn radius. The heading angle is estimated as the angle difference between consecutive points on the path, while the heading rate angle is determined with a three-point finite differencing method of the heading angle vector. The time is estimated from the path solution given the constant speed of the SUAS. This path solution will be used as an initial guess for the state, control, and time of the two-dimensional SUAS optimal control problem to acquire the optimal path through the defined CSC.

To end the path on an orbit around the target, first, the orbit radius must not intersect a constrained edge and, second, an estimated terminal point must be determined for the initial guess vector. To assure that the desired orbit radius around the target remains in feasible space, a line intersection algorithm is used to determine the distance from the target location to the closest constrained edge, providing an upper bound to the orbit radius. This is determined by finding the perpendicular distance from the target point to each constrained line segment. Constructing Bourke's algorithm [28], the slope λ of the perpendicular line segment is defined as

$$\lambda = \frac{(x_t - x_1)(x_2 - x_1) + (y_t - y_1)(y_2 - y_1)}{\|(x_2, y_2) - (x_1, y_1)\|_2^2} \tag{5}$$

where (x_t, y_t) defines the target location and (x_1, y_1), (x_2, y_2) define the vertex points of the constrained edge. A line equation is used to determine the point of intersection between the perpendicular line and the constrained line vector,

$$x_3 = x_1 + \lambda(x_2 - x_1) \tag{6}$$

$$y_3 = y_1 + \lambda(y_2 - y_1) \tag{7}$$

The shortest distance from the target point to the constrained edge, ζ_k, is defined by

$$\zeta_k = \min(\|(x_t, y_t) - (x_i, y_i)\|_2) \qquad \forall\ i \in [1, 2, 3] \tag{8}$$

Evaluating Eq. (8) for each constrained edge, the maximum orbit radius for the SUAS is defined by the minimum distance to the closest constrained edge in the domain,

$$R_{\max} = \min(\zeta_k) \qquad \forall\ k \in [1 \ldots C] \tag{9}$$

where C defines the total number of constrained edges. This process is shown in Fig. 5 for a single constrained edge.

Finally, the $R_{\max}$ value is checked against the required orbit radius R_t to assure that the SUAS remains in an unconstrained, feasible airspace while accomplishing its mission. The final orbit radius is then used to determine the intersection of simplexes, which are removed from the search corridor for the terminal condition. This

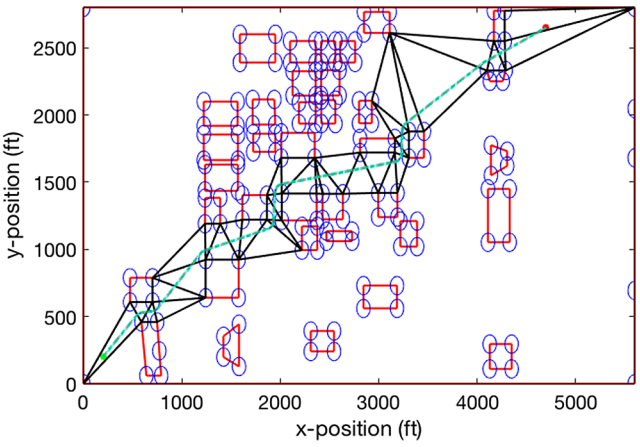

Fig. 4 Triplanner connected simplex channel and path solution.

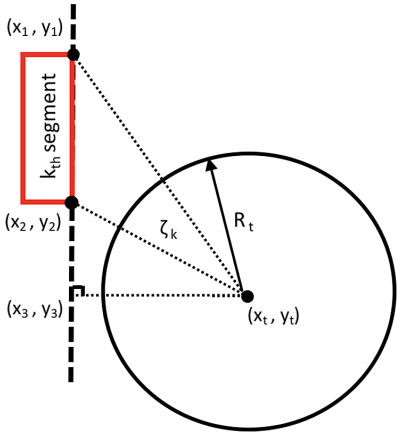

Fig. 5 Maximum orbit radius.

allows for the NLP solver to converge to an optimal solution without the requirement of traversing through a specific simplex order in the final phase of the path.

The initial guess from the final simplex phase through the intersection of the target orbit consists of a straight-line solution to the intersection of the orbit radius. This line is determined by defining the unit vector from the final position of the last simplex of the Triplanner solution to the target point. A line intersection algorithm is then used to find the transition point from the straight line path to a circular orbit based on the length of the target radius and the vehicles minimum turning radius.

B. Optimal Control Problem

Within the construct of a CSC, the optimal control problem is formulated into a phased approach. Each simplex that is traversed is represented as a single phase p, and each phase is connected through event constraints equating the states, control, and time at each boundary. The total number of phases is represented as P. The CSC is defined through the results of the Triplanner algorithm and account for phases 1: $N-1$. The final phase is evaluated in global coordinates to allow for left or right turn orbits of varying radius around the target. The SUAS is defined with two-dimensional dynamics as

$$\dot{x}(t) = v\cos(\theta(t)) \tag{10}$$

$$\dot{y}(t) = v\sin(\theta(t)) \tag{11}$$

Here, the velocity v is held constant at 30 ft/s through the entirety of this work. The state vector, given Cartesian coordinates, is defined as

$$\tilde{X} = (x, y, \theta) \tag{12}$$

with control

$$u = \dot{\theta} \tag{13}$$

Transforming the coordinates into a barycentric coordinate frame and taking the derivative of the weight equations with respect to the x and y coordinates, a new set of dynamic equations can be defined to propagate the state variables through each simplex. This results in an equivalent set of dynamics defined through the barycentric weights, previously described in [6] as

$$\dot{\alpha}_1^{(p)}(t) = \frac{(y_2 - y_3)(v)\dot{x}(t) + (x_3 - x_2)(v)\dot{y}(t)}{\det(T)} \tag{14}$$

$$\dot{\alpha}_2^{(p)}(t) = \frac{(y_3 - y_1)(v)\dot{x}(t) + (x_1 - x_3)(v)\dot{y}(t)}{\det(T)} \tag{15}$$

$$\dot{\alpha}_3^{(p)}(t) = -\dot{\alpha}_1^{(p)}(t) - \dot{\alpha}_2^{(p)}(t) \tag{16}$$

$\forall\ p \in [1 \ldots P-1]$, where T is a 2×2 matrix representing the vertices of the triangle as

$$T = \begin{pmatrix} x_1 - x_3 & x_2 - x_3 \\ y_1 - y_3 & y_2 - y_3 \end{pmatrix} \tag{17}$$

the control consistent with Eq. (13), and the new state vector, given in barycentric coordinates, defined as

$$X = (\alpha_1, \alpha_2, \alpha_3, \theta) \tag{18}$$

With the dynamics described in terms of a weighted measure to each simplex's vertex, a phased solution can be constructed where each simplex represents a single phase in the optimal control problem.

By transforming the problem to the barycentric coordinate frame, parameter bounds on the state can be generalized for each phase by taking advantage of the coordinate frame properties defined as

$$0 \le \alpha_1^{(p)}, \alpha_2^{(p)}, \alpha_3^{(p)} \le 1 \tag{19}$$

Parameter bounds on the heading, control, and time are given as

$$|\theta^{(p)}| \le 180 \ \text{deg} \tag{20}$$

$$|u^{(p)}| \le 25 \ \text{deg/s} \tag{21}$$

$$0 \le t^{(p)} \le \frac{(\rho) * \text{edge}_{\max}^{(p)}}{v} \tag{22}$$

where $\text{edge}_{\max}$ represents the longest edge in the defined simplex and $\rho > 1$ defines a weighting to increase the time bound to allow for curved solutions through a simplex, accounting for potential flight through keep-out regions. Bounds for the final phase are determined by the maximum and minimum values of the final simplex edge and the orbit around the target.

Finally, event constraints are imposed to maintain a continuous transition of the state variables through each simplex,

$$X_o^{(p)} - X_f^{(p-1)} = 0 \qquad \forall\ p \in [2 \ldots P] \tag{23}$$

The final phase requires the vehicle to end on a constant-radius orbit around the target. This phase takes place outside of the CSC requiring the implemented dynamics to be in global Cartesian coordinates represented in Eqs. (10) and (11) with control as the heading rate, $\dot{\theta}$. The event constraint relating phase $P-1$ to P must be equated through a transformation of the coordinate states of the $P-1$ phase as follows:

$$x_f^{(P-1)} = \alpha_{1f}^{(P-1)} x_1 + \alpha_{2f}^{(P-1)} x_2 + \alpha_{3f}^{(P-1)} x3 \tag{24}$$

$$y_f^{(P-1)} = \alpha_{1f}^{(P-1)} y_1 + \alpha_{2f}^{(P-1)} y_2 + \alpha_{3f}^{(P-1)} y3 \tag{25}$$

where x_i and y_i represent the simplex vertex locations. This results in the event constraint

$$\tilde{X}_o^{(P)} - \tilde{X}_f^{(P-1)} = 0 \tag{26}$$

This constraint bridges the state and control of the final simplex edge to the final phase. To account for the terminal condition, a final event constraint is added to implement the tangency condition to the final orbit. The tangency condition is defined for the final x and y positions as

$$\tilde{x}_f = x_t + R_t \cos\left(\theta_f + \mu\frac{\pi}{2}\right) \tag{27}$$

$$\tilde{y}_f = y_t + R_t \sin\left(\theta_f + \mu\frac{\pi}{2}\right) \tag{28}$$

where $\tilde{x}_f$ and $\tilde{y}_f$ represent a vehicle position coincident and perpendicular to the desired final orbit of center x_t, y_t with radius R_t, while $\mu \in [-1, 1]$ implements a clockwise or counterclockwise orbit dependent on the vehicles mission requirements. The final event constraint is applied to the first two states described as

$$\tilde{x}_f - x_f^P = 0 \tag{29}$$

$$\tilde{y}_f - y_f^P = 0 \tag{30}$$

With the dynamics and parameter bounds defined, the objective function is designed for minimum time of flight, while including penalties for incursions into keep-out regions. The cost associated with the minimum time flight through each phase is represented as

$$J^{(p)}_{\min T} = \int_{t_0^{(p)}}^{t_f^{(p)}} \mathrm{d}t \qquad \forall\ p \in [1 \ldots P] \tag{31}$$

The penalty for the keep-out regions, $F_i(x, y)$, is defined with a sigmoid function, $\phi(F)$, and minimizes the keep-out incursions at each collocation point as

$$\phi_i(F_i) = \frac{1}{1 + e^{(s_i(F_i(x,y)-1)}} \tag{32}$$

for

$$F_i(x, y) = \left(\frac{(x^{(p)}(t) - Kx_i)}{a_i}\right)^2 + \left(\frac{(y^{(p)}(t) - Ky_i)}{b_i}\right)^2 \tag{33}$$

where Kx_i and Ky_i define the keep-out center point, and a_i and b_i define the semi-major and semi-minor axes. This yields a cost function, minimizing the incursion onto keep-out regions, defined as

$$J^{(p)}_{\min E} = \int_{t_0^{(p)}}^{t_f^{(p)}} \phi_i^{(p)}(F_i)\,\mathrm{d}t \qquad \forall\ p \in [1 \ldots P-1] \tag{34}$$

The complete objective function is a summation of the minimum time cost and minimum incursion to keep-out regions.

$$J = \sum_{p=1}^{P} \beta J^{(p)}_{\min T} + \sum_{p=1}^{P-1} (1-\beta) J^{(p)}_{\min E} \tag{35}$$

Here, β defines the weighting on the components of the cost, such that the cost function influences the desired flight path. With the cost function defined, integral bounds are implemented within the optimal control problem as

$$0 \le J_{\min E} \le 50 \tag{36}$$

The optimal control problem is solved using the General Purpose Optimization Pseudospectral Software (GPOPS-II). The parameters used herein are shown in Table 1.

IV. Scenario

A city map representing an urban environment is considered, where an aircraft is required to maintain constant altitude and fly from an initial position to a target of interest. This scenario is represented with a constraint map of downtown Chicago, USA. The SUAS is required to maintain an altitude of 600 ft AGL. Each building that exceeds 550 ft AGL is modeled as a constraint that must be avoided. To allow for variations in the aircraft position due to exogenous inputs during flight, the most restrictive building positions are modeled between the altitude slice of 550 and 650 ft, which includes a 15 ft safety buffer from each vertex point to assure that the SUAS maintains a safe distance between any structure. Initially, the minimum time flight path is determined without modeling keep-out regions in order to illustrate the accuracy and computational efficiencies of the algorithm. The initial path generated for this solution was shown previously in Fig. 4.

Table 1 GPOPS-II user settings

Mesh method	hp-PattersonRao
Mesh tolerance	10^{-2}
NLP solver	SNOPT
Derivative supplier	SparseCD
Method	RPM-differential
NLP tolerance	10^{-5}
Min collocation points	4
Max collocation points	10
Mesh fraction	1/2*ones(1,2)
Mesh collocation points	4*ones(1,4)

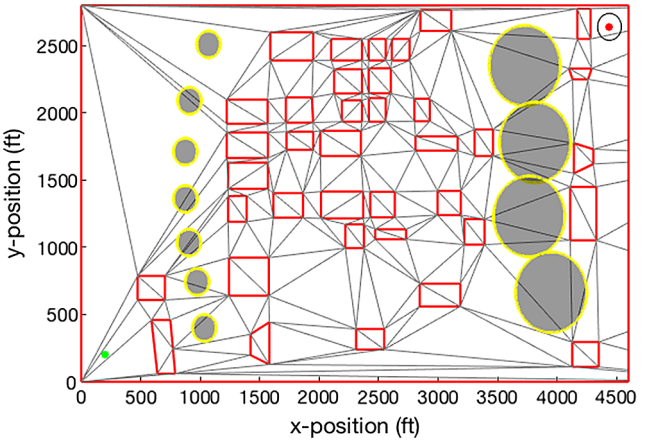

Fig. 6 Downtown Chicago Constraint Map. [Map Data @2017 Google.]

This scenario is expanded to minimize incursions to included keep-out regions. The vehicle is now required to fly from an initial position to a final region, culminating on an orbit around a target of interest, while keep-out regions representing minimal flight zones are included along the west river and down State Street. These zones must be avoided when possible. Figure 6 illustrates the constraint map for the intended scenario as well as the triangulated mesh over the domain. The initial starting position and the target location are shown with the asterisks in the bottom left and top right corners of the map, respectively. The four-sided polygons outline building constraints that must be avoided, whereas the gray keep-out regions are modeled as circular regions upon which the SUAS must minimize incursions. The left column of keep-out regions has a separation between them, allowing the aircraft to fully avoid the minimal flight zones if they reside completely inside the triangulated search corridor. The right column of keep-out regions maintain a small overlap, requiring flight through the zones at a minimum incursion level, dependent on the CSC boundary. Other locations and sizes for keep-out regions may require the aircraft to fly through the center of undesirable airspace. If a keep-out region is unavoidable, an analysis on the region should be conducted and portions of the region should be modeled as a hard constrained and removed from the search space if warranted. The circle around the target point, located at (4440, 2640) ft represents the desired orbit for the SUAS at $R_t = 100$ ft.

V. Results

In the first solution, without including keep-out regions, an initial guess vector for the state, control, and time is determined for each simplex through the Triplanner toolkit solution and implemented in GPOPS-II as individual phases. All simulations were conducted with GPOPS-II in MATLAB version R2016b. All results were acquired on a 2016 iMac with 2.8 GHz quad-core Intel i5 processor and 8 GB of 1867 MHz LPDDR3 memory. The results are shown in Fig. 7.

The initial guess is illustrated with the dotted line, whereas the discretized path solution is shown with a series of asterisks representing the LGR points of the solution. An optimal solution through the search corridor was found with a computation time of 2.1 s. The difference between the optimal path solution and the Dubins path acquired from Triplanner is found in the location of the aircraft's turn points and highlighted in the expansion box of Fig. 7. The minimum radius turns for the Dubins path are located on the vertex of each constraint encountered along the path, whereas the optimal path determines the most efficient turn point between two constraints. This allows for 5.1% reduction in path length, decreasing the total

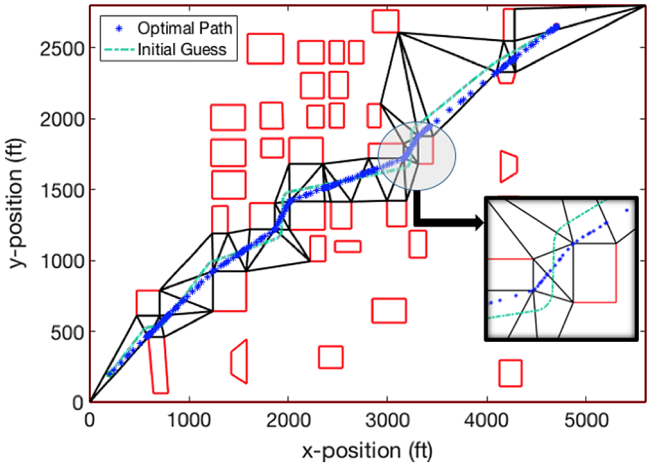

Fig. 7 Path solution.

path length from 5491 ft with the Triplanner solution to 5213 ft with the optimal solution. Additionally, the optimal solution provides the heading angle and heading angle rates as shown in Fig. 8. The initial guess for these vectors is derived from the Triplanner toolkit solution shown with the dotted line and the optimal heading angle and control shown with the solid line.

Including keep-out regions, the initial guess used to seed the NLP is again determined using the Triplanner algorithm. The output of Triplanner, shown in Fig. 9, consists of a Dubins path solution contained inside a CSC shown as a series of black simplexes. All building constraints are contained outside of the search corridor, thus eliminating path constraints from the problem formulation. The initial

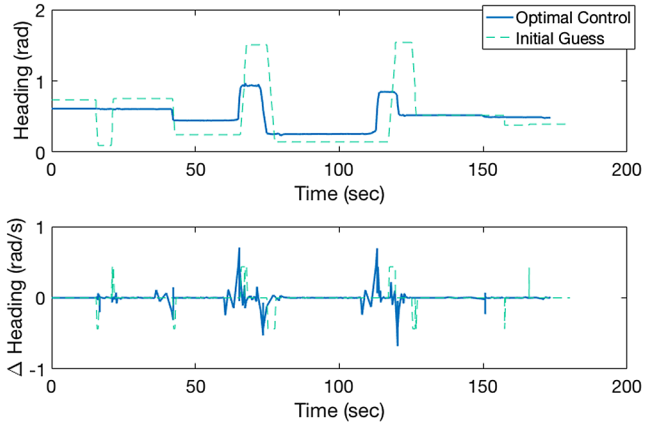

Fig. 8 Path solution.

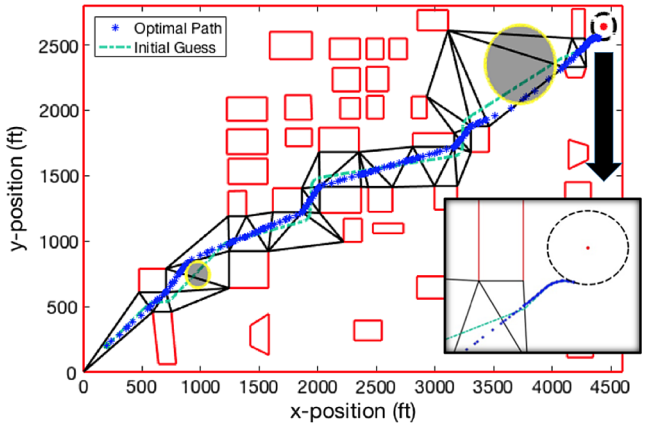

Fig. 9 Keep-out region path solution.

path guess found by Triplanner, shown with the dashed line, remains unchanged from the first scenario as the geometric path planner is unable to account for keep-out regions where minimal incursions are allowed; however, it now includes terminating the path on a defined orbit radius. Note that only the simplexes in the CSC, as well as the sigmoid functions modeling the keep-out regions, are presented to the NLP solver so that a computationally efficient search can be performed for the optimal solution within the defined corridor. The optimal solution is shown within the CSC with the asterisks defining the LGR points of the solution.

The asterisks shown define the LGR collocation points of the optimal path through the triangulated search corridor. The first keep-out region is fully avoided because the region is located completely within the CSC. The second keep-out region extends beyond both boundaries of the CSC, and therefore the path is dependent on the weighted cost function, set to $\beta = 0.7$ in this work. The optimal path chosen avoids the keep-out region up to the limit of the triangulated search corridor. The final terminal condition is achieved with the SUAS ending on an orbit around the final target. This path was achieved in 95.5 s computation time. The computation time for this scenario was significantly increased compared with the scenario without keep-out regions. This increase is attributed to the gradient-based constraint functions defining the keep-out regions as well as the terminal condition constraint. Because of the nature of the global solution of direct orthogonal collocation techniques, the computation time per phase of the solution is unknown. Ultimately, this methodology allows for path solutions when terminal constraint functions or minimal incursions to keep-out regions are required by the mission objectives.

The vehicle heading and control vectors are shown in the top two plots of Fig. 10. The third plot shows the integrated value of the cost function of Eq. (34) evaluated over each phase.

The deviation in the heading for the first keep-out region can be seen in the heading vector between 20 and 40 s into the flight. The lower heading value at the 70 s mark illustrates the benefits gained from the optimal solution over the Dubins path solution. The second keep-out region is illustrated at the 105 s mark of the first plot, and the final phase can be seen as a minimum radius to a tangent location on the final orbit, shown at 160 s into the simulation. The integrated value of the sigmoid function on the third plot shows the impact the two keep-out regions have on the optimal control solution. The integrated values are referenced in each phase rather than by flight time as the cost associated with the sigmoid function corresponds to individually defined simplexes and the time associated with each simplex is not consistent, as the size of the simplexes is dependent on the discretized mesh and therefore not equally spaced. The first keep-out region only slightly affected the cost function as seen in phases 6

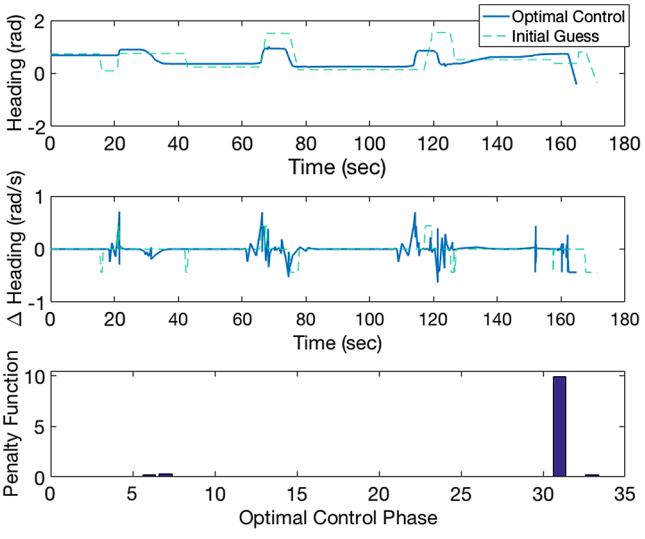

Fig. 10 Optimal solution state and control.

and 7. The second keep-out region is shown to have a greater effect on the cost function as the region is unavoidable, affecting phases 30 through 33. The value of the constraint portion of this cost can be adjusted through the stiffness parameter of the sigmoid function described previously. For this work, $s = 2$ and $\beta = 0.7$ were chosen to acquire an adequate solution. Adjusting these parameters will change the optimal path solution based on the performance desired. Increasing the stiffness parameter, s, allows the flight path to further intrude on the keep-out region as shown in Fig. 1. Similarly, increasing the weight in the objective function will decrease the influence of the sigmoid function on the objective, accomplishing the same result as increasing the stiffness parameter but with less sensitivity to the results. Ultimately, the choice of these values is based on the perceived influence the keep-out region will have on the aircraft for the task presented.

VI. Conclusions

This work illustrated a methodology to solve the optimal control path planning problem for small unmanned aircraft systems (SUAS) in highly constrained environments. Without a simplex approach, an initial guess for a solution would be difficult to acquire and an optimal control solver would be burdened with a large number of constraint functions, inhibiting a path solution from being found. With a Connected Simplex Corridor (CSC) construct implemented, the Triplanner algorithm provides a computationally efficient algorithm where a quality guess is used to seed the nonlinear program in a phased approach, thus eliminating the 37 building constraints from the search domain. In designing the optimal control problem, constraints must be evaluated to determine if they consist of a flight region for which incursions can be minimized, or if they instead represent a region that must be avoided completely. If the constraint must be completely avoided, the Triplanner algorithm's triangulated mesh will remove the constraint from the search field, limiting the problem's domain. A conservative approach to implementing constraints, combined with the minimum turning radius of the SUAS, could result in Triplanner failing to return a feasible path solution due to an overconstrained environment. If, however, constraints are modeled as keep-out regions where the SUAS must minimize time spent in a flight zone, it was demonstrated herein that the constraint should be modeled in the optimal control problem rather through mesh adjustments in the geometric path planner. By modeling the constraint in the cost function, tuning parameters can be used for varying levels of incursions within the keep-out region based on the defined CSC and the risk of designing an overconstrained problem is minimized.

The sigmoid function was shown to be a viable option for modeling constraints in the optimal control cost function within the construct of a CSC. These functions are appropriate for gradient-based optimization software as they provide smooth, bounded, and differentiable functions. The final cost function was a weighted sum, distributing the cost over flight time and time within keep-out regions. These weights, along with the stiffness parameter of the sigmoid function, can be tuned to achieve desired results based on the level of incursion permitted within the keep-out regions.

Ultimately, a CSC was used to provide a foundation for optimal control solutions. When paired with direct orthogonal collocation methods for optimal control, both hard constraints, such as buildings and terrain, as well as keep-out regions, such as unavoidable flight zones, can be modeled and optimal path solutions can be attained. Significant to the work is the published computation times for nonlinear, gradient-based path constraints modeled in a highly constrained urban environment, and thus continuing to close the gap to real-time, on-board optimal path planning solutions.

Acknowledgments

The authors would like to thank the Air Force Research Laboratory, Aerospace Systems Directorate, Power and Controls Division, who sponsored the work and provided the challenging problem concept, background information, and continual support. Previous version of this paper was presented at the American Controls Conference, Milwaukee, WI, 27–29 June 2018.

References

[1] Air Force Research Laboratory, "Air Force Research Laboratory Autonomy Science and Technology Strategy," Tech. Rept., Air Force Research Lab., U.S. Air Force, Wright-Patterson AFB, OH, 2013.

[2] Smith, N. E., "Optimal Collision Avoidance Trajectories for Unmanned/Remotely Piloted Aircraft," Ph.D. Dissertation, Air Force Inst. of Technology, Wright-Patterson AFB, OH, 2014.

[3] Humphreys, C. J., and Cobb, R. G., "A Hybrid Optimization Technique Applied to the Intermediate-Target Optimal Control Problem," *Global Journal of Technology and Optimization*, Vol. 7, No. 2, 2016, pp. 1–8. doi:10.4172/2229-8711

[4] Masternak, T. J., "Multi-Objective Trajectory Optimization of a Hypersonic Reconnaissance Vehicle with Temperature Constraints," Ph.D. Dissertation, Air Force Inst. of Technology, 2014.

[5] Carr, R. W., and Cobb, R., "An Energy Based Objective for Solving an Optimal Missile Evasion Problem," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2017-1016, 2017, pp. 1–18. doi:10.2514/6.2017-1016

[6] Miller, J. K., Llanos, P. J., and Hintz, G. R., "Optimal Control Framework for Impulsive Missile Interception Guidance," *Guidance, Navigation, and Control Conference*, AIAA Paper 2014-4787, 2014, pp. 1–14. doi:10.2514/6.2013-4787

[7] Betts, J. T., "Practical Methods for Optimal Control and Estimation Using Nonlinear Programming," *Advances in Design and Control*, Soc. for Industrial and Applied Mathematics, Philadelphia, PA, 2008, p. 434.

[8] Suplisson, A. W., "Optimal Recovery Trajectories for Automatic Ground Collision Avoidance Systems (Auto GCAS)," Ph.D. Dissertation, Air Force Inst. of Technology, Wright-Patterson Air Force Base, OH, 2015.

[9] Garg, D., "Advances in Global Psuedospectral Methods for Optimal Control," Ph.D. Dissertation, Univ. of Florida, Gainesville, FL, 2011.

[10] Burden, R. L., and Faires, J. D., *Numerical Analysis*, 9th ed., Brooks/Cole, Aug. 2010, pp. 173–202.

[11] Benson, D., Huntington, G., Thorvaldsen, T., and Rao, A., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1435–1440. doi:10.2514/1.20478

[12] Humphreys, C. J., Cobb, R. G., Jacques, D. R., and Reeger, J. A., "Dynamic Re-Plan of the Loyal Wingman Optimal Control Problem in a Changing Mission Environment," *AIAA Infotech @ Aerospace*, AIAA Paper 2017-0746, 2017, pp. 1–13. doi:10.2514/6.2016-0746

[13] Carr, R. W., and Lagimoniere, E., "A Range Safety Footprint Analysis for the Dream Chaser Engineering Test Article Using Trajectory Optimization," *AIAA Guidance, Navigation, and Control (GNC) Conference*, AIAA Paper 2013-4647, 2013. doi:10.2514/6.2013-4647

[14] Humphreys, C. J., Cobb, R. G., Jacques, D. R., and Reeger, J. A., "Dynamic Re-Plan of the Loyal Wingman Optimal Control Problem in a Changing Mission Environment," *AIAA Infotech @ Aerospace* , AIAA Paper 2016-0746, 2016, pp. 1–15. doi:10.2514/6.2016-0746

[15] Zollars, M. D., and Cobb, R. G., "Simplex Methods for Optimal Control of Unmanned Aircraft Flight Trajectories," *ASME Dynamics Systems and Controls Conference*, Paper DSCC2017-5031, Tysons Corner, VA, 2017, p. 10. doi:10.1115/DSCC2017-5031

[16] Myers, D., Batta, R., and Karwan, M., "A Real-Time Network Approach for Including Obstacles and Flight Dynamics in UAV Route Planning," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, Vol. 13, No. 3, 2016, pp. 291–306. doi:10.1177/1548512916630183

[17] Aoude, G. S., "Two-Stage Path Planning Approach for Designing Multiple Spacecraft Reconfiguration Maneuvers and Application to SPHERES Onboard ISS," S.M. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, 2007.

[18] Aoude, G. S., How, J. P., and Garcia, I. M., "Two-Stage Path Planning Approach for Solving Multiple Spacecraft Reconfiguration Maneuvers," *The Journal of the Astronautical Sciences*, Vol. 56, No. 4, 2008, pp. 515–544. doi:10.1007/BF03256564

[19] Kapadia, M., and Badler, N. I., "Navigation and Steering for Autonomous Virtual Humans," *Wiley Interdisciplinary Reviews: Cognitive*

Science, Vol. 4, No. 3, 2013, pp. 263–272.
doi:10.1002/wcs.1223

[20] Kallmann, M., "Shortest Paths with Arbitrary Clearance from Navigation Meshes," *Proceedings of the Eurographics SIGGRAPH Symposium on Computer Animation SCA*, Madrid, Spain, 2010, pp. 159–168.

[21] Kallmann, M., "Dynamic and Robust Local Clearance Triangulations," *ACM Transactions on Graphics*, Vol. 33, No. 5, 2014, p. 1–17.
doi:10.1145/2672594

[22] Kallmann, M., "Navigation Queries from Triangular Meshes," *Motion in Games*, edited by R. Boulic, Y. Chrysanthou, and T. Komura, Lecture Notes in Computer Science, Vol. 6459, Springer, Berlin, 2010, pp. 230–241.

[23] Zollars, M. D., Cobb, R. G., and Grymin, D. J., "Simplex Optimal Control Methods for Urban Environment Path Planning," *2018 AIAA Information Systems—AIAA Infotech @ Aerospace*, AIAA Paper 2018-2259, 2018, p. 16.
doi:10.2514/6.2018-2259

[24] Zollars, M. D., Cobb, R. G., and Grymin, D. J., "Simplex Solutions for Optimal Control Flight Paths in Urban Environments," *Journal of Aeronautics and Aerospace Engineering*, Vol. 6, No. 3, 2017, p. 8.
doi:10.4172/2168-9792.1000197

[25] Zollars, M. D., Cobb, R. G., and Grymin, D. J., "Optimal Path Planning for SUAS Waypoint Following in Urban Environments," *IEEE Aerospace Conference*, IEEE Publ., Piscataway, NJ, 2018, p. 10.
doi:10.1109/AERO.2018.8396483

[26] Dechter, R., and Pearl, J., "Generalized Best-First Search Strategies and the Optimality of A*," *Journal of the ACM*, Vol. 32, No. 3, 1985, pp. 505–536.
doi:10.1145/3828.3830

[27] Hershberger, J., and Snoeyink, J., "Computing Minimum Length Paths of a Given Homotopy Class," *Computational Geometry: Theory and Applications*, Vol. 4, No. 2, 1994, pp. 63–97.
doi:10.1016/0925-7721(94)90010-8

[28] Bourke, P., "Points, Lines, and Planes," 1988, http://paulbourke.net/geometry/pointlineplane/ [retrieved 10 Aug. 2017].

J. Krozel
Associate Editor