2016-6

UAV Path-Planning using Bézier Curves and a Receding Horizon Approach

Bryce Ingersoll
Brigham Young University, bryceingersoll@gmail.com

Kyle Ingersoll
Brigham Young University

See next page for additional authors

Original Publication Citation
Ingersoll, B., Ingersoll, K., DeFranco, P., and Ning, A., "UAV Path-Planning using Bézier Curves and a Receding Horizon Approach," AIAA Modeling and Simulation Technologies Conference, Washington, DC, Jun. 2016. doi:10.2514/6.2016-3675

UAV Path-Planning using Bézier Curves and a Receding Horizon Approach

Bryce Ingersoll,* Kyle Ingersoll,† Patrick DeFranco,‡ and Andrew Ning§
Brigham Young University, Provo, Utah, 84602

Unmanned aerial vehicles (UAVs) are used in an increasing number of applications. Such applications may include navigating through heavy traffic and highly congested airways, where numerous static and dynamic obstacles impinge upon a UAV's flight. It is imperative that a UAV successfully avoids these obstacles, while improving its planned flight path according to certain criteria. We have modeled UAV path planning as a single objective optimization problem that utilizes a receding horizon approach, where the path is constrained to avoid obstacle collision and to account for flight aerodynamic constraints. The proposed method is gradient based, allowing for quick and robust convergence to a near optimal solution. This heuristic method converges closely to full-knowledge optimal solutions and will allow UAVs to be implemented in a greater amount of tasks and missions than before while lessening the risk to the safety of others and the safety of the UAV.

Nomenclature

i	index for path segments
j	index for obstacles
m	number of planned path segments
n	number of static obstacles
r	radius, m
t	Bézier curve parametrization value
$\vec{v}$	dynamic obstacle speed, m/s
V	UAV velocity, m/s
x, y	coordinates in domain
ℓ	path length, m
ℓ_f	distance between UAV and final destination, m

I. Introduction

A. Background

Unmanned aerial vehicles (UAVs) are used in an increasing number of applications, such as delivery, forestry and environmental assessments, security, and other types of surveillance. While on a mission, it is vital that UAVs detect and avoids obstacles. Collisions between UAVs and objects not only result in property damage, but also present potential harm to bystanders. These obstacles can both be static, such as mountains and buildings, and dynamic, such as other aircraft. In addition to obstacle avoidance, UAVs need to optimize their proposed flight path. This is essential so that UAVs can perform their missions efficiently. This can be measured in a number of ways, such as flight path length, time elapsed to complete the mission, or energy use.

*Undergraduate Research Assistant, Department of Mechanical Engineering, AIAA Student Member
†Graduate Student, Department of Mechanical Engineering
‡Graduate Student, Department of Electrical Engineering
§Assistant Professor, Department of Mechanical Engineering, AIAA Senior Member

Several UAV path planning methods have already been proposed. One commonly cited method, the Voronoi graph,[1] models each obstacle as a point and then partitions the area into a set of convex cells that each contain only one obstacle. The wall of each cell is closest to the obstacle contained in that cell compared to other obstacles. When applied to path planning, the edges of the cells can be traced to obtain a path through the obstacle field. Two immediately apparent shortcomings of this method are an inability to model obstacles with nonzero area and an inability to produce non-smooth paths. Another commonly cited method, rapidly exploring random trees[2] (RRT), is an exploration algorithm that uniformly, but randomly, searches an area. When using RRTs for path planning, the algorithm checks for a feasible path from the branch ends to the final destination at the end of each iteration. When a feasible path is found, the algorithm then traverses the path and searches for possible links between non-consecutive path nodes, thus smoothing and shortening the path. The RRT method also produces non-smooth paths that are inherently difficult for a UAV to precisely follow.

Another common group of UAV path planning methods use genetic, or evolutionary, algorithms. These gradient-free approaches combine steps of selection, crossover, and mutation to search the domain for the optimal path.[3] One approach using a genetic algorithm models 3D UAV flight with a variety of mission objectives.[4] Evolutionary algorithms have been implemented for their high robustness, ease of implementation, and their high adaptability.[5] However, genetic algorithms are inefficient in converging to an optimal solution.[6] Gradient-based approaches have also been implemented which converge more quickly without loss of robustness.[7] In addition, a number of heuristic approaches have been developed for situations where there are many degrees of freedom[8,9] and a partial knowledge of the obstacle field.[10]

The actual planned path of the UAV has been modeled in a number of ways. One method[11] uses a combination of straight lines and constant radius arcs to model the path. Another approach[12] adds cothoid arcs to this combination. Dubins curves have also been used.[13] A common method of modeling UAV flight paths are Ferguson splines, which are implemented in a number of different approaches.[3,14,15] Ferguson splines are frequently used because they allow the position and derivative at the beginning and end of the spline to be set, independent of the other parameters. B-splines are also commonly used[4,16] because of the ease to increase or decrease the order of the spline.

Finally, there are a number of ways that obstacle avoidance has been approached. One method adds a penalty to its objective function as its planned path nears an obstacle,[14] while another approach imposes a penalty only when the planned path would result in a collision.[15] Obstacles can also be treated as flight restricted areas[17] in lieu of a single point. While many approaches assume complete knowledge of a static obstacle field, there are also approaches which account for uncertain obstacle position in dynamic environments.[11,18] In addition, many approaches account for wind patterns and other environmental effects.[19–21]

A myriad of UAV path planning algorithms have been proposed. However, there is need for a reliable and robust algorithm that can operate within a limited field of view. To work in real time, this method would need to be gradient based to allow for quick convergence for a given tolerance. This method would also need to improve a flight path with respect to various criteria while avoiding numerous static and dynamic obstacles. Our objective is to create such an approach, which would be valuable in certain situations, such as urban or forest environments, and would allow UAVs to be implemented in a greater number of applications.

B. Motivation

In this paper we model UAV flight path planning as a constrained single objective optimization problem. The optimization approach attempts to minimize a single performance criteria, such as flight path length, time elapsed to complete the mission, or energy use. A receding horizon approach[10] is used, where an incomplete knowledge of the obstacle field is assumed. Because of this lack of knowledge, an optimal solution for the entire domain is not found, but rather an optimal solution within the horizon of information. This approach, as will be shown, converges closely to full-knowledge optimal solutions. The approach is gradient based, allowing for quick convergence, and it accounts for numerous static and dynamic obstacles. Such an approach will allow UAVs to be implemented in a greater amount of tasks and missions than before. For instance, when a UAV is tasked to travel through an urban environment it must quickly navigate around numerous obstacles while avoiding collision. A similar situation can occur in a forested environment when the UAV travels underneath the tree line to avoid detection. In both situations, the UAV needs to respond quickly to avoid nearby obstacles, while planning a flight path which is improved according to some efficiency criteria. Our proposed method will be able to accomplish both objectives. Note that we assume the UAV is able to detect the obstacle in its field of view, so this approach focuses only on avoidance.

II. Methodology

We modeled the UAV path planning problem as a constrained, single objective optimization problem. Our objective was to minimize some function f with respect to a number of design variables with both equality and inequality constraints. Section A details our approach and initialization of the problem. Section B describes the design variables used, Section C the objective functions, and Sections D the constraints. Finally, Section E details the multi-start approach used.

A. Optimization Approach

Instead of planning out the entire path, a receding horizon approach was implemented to plan segments as the path marched towards the final destination. This modeled the UAV's limited field of view; it only had knowledge of obstacles near its position, and thus only planned to avoid those obstacles. Additionally, such an approach reduced computation time making the optimization problem more manageable. We assumed that each path planned was associated with a change in time Δ_{time}, which was necessary to define the time and energy use objective functions and certain flight constraints, as well as the UAV speed. Each segment was planned by solving for multiple segments which each occurred consecutively, as seen in Figure 1. The blue circles represent static obstacles which the UAV must avoid. The UAV is represented as a yellow/blue line; the line is blue when the UAV is traveling at its minimum velocity and transitions to yellow when it flies at its maximum velocity. The thin yellow/blue line represents path segments which have previously been planned, and the bold yellow/blue line represents the current segment being planned. In this case, two path segments are used ahead of the current segment (represented by the yellow/blue dashed segments). The UAV is not committed to travel these segments; they are used for planning obstacle avoidance.

To initialize the problem, we specified the field of flight as a 2D 100 x 100 domain and set the UAV's initial position as $((x_0, y_0) = (0, 0))$ and final destination as $((x_f, y_f) = (100, 100))$. We specified the number of static obstacles to n, where $40 \leq n \leq 60$. For each obstacle field, the obstacles' position were randomly placed within $(5, 5) \leq (x_j, y_j) \leq (95, 95)$ in the domain. A circular uncertainty bound with radii ranging from $r_{min} = 4$ to $r_{max} = 7$ was randomly set around each obstacle. Finally, we specified the initial position, size, trajectory, and number of dynamic obstacles. These parameters vary to demonstrate different applications of this method, and are specified in Section III. Dynamic obstacles were colored dark red to differentiate from static obstacles in blue.

We used MATLAB's *fmincon* solver for this approach. Its *algorithm* setting was set to *sqp*, and its maximum function evaluations and iterations were set large enough to never be active. Analytic gradients were calculated using the complex step method.

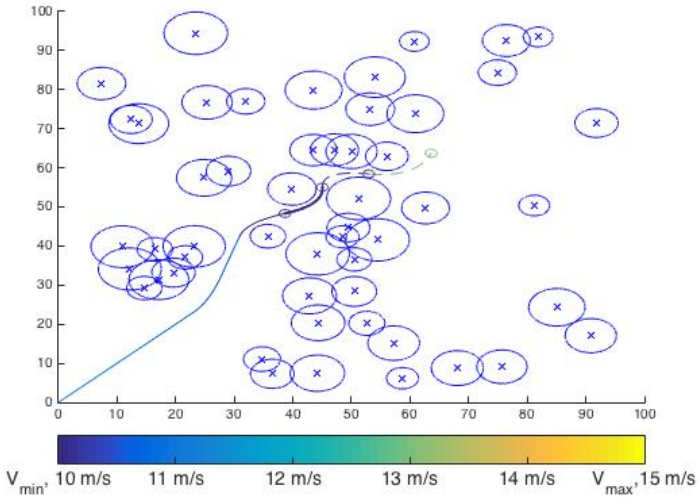


Figure 1: An example of the path planning approach. The UAV's path (yellow/blue) successfully avoids the static obstacles (blue).

B. Design Variables

Similar to other approaches,[4] quadratic Bézier curves[22] were used to model the UAV path, which are defined as:

$$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2,\ 0 \leq t \leq 1 \tag{1}$$

Bézier curves were used for several reasons. Its first derivative at any point on the curve was easily computed using:

$$B'(t) = 2(1-t)(P_1 - P_0) + 2t(P_2 - P_1) \tag{2}$$

This made calculating certain flight constraints, such as minimum turn radius, very straightforward. We also used Bézier curves because of their simplicity. While other approaches use more complex splines, using Bézier curves allowed for quick computation of curvature and reduced the number of design variables.

For each path segment, the first point was set as the last point of the previous segment, or $P_{0(i)} = P_{2(i-1)}$. In the case of the initial path segment being planned, $P_0 = (x_0, y_0)$. Every point contains an x and y coordinate, so four design variables were used for each segment planned. Thus, for each path planning iteration, there were $4m$ design variables. For example, when three segments were being planned, a total of 12 design variables were used.

C. Objective Functions

UAVs are tasked with a variety of missions with different objectives, so we defined three separate functions which evaluated different quantities to be minimized: path length, time elapsed when traveling the planned path, and energy use when traveling the path. Because this method models a receding horizon approach, we are only certain about these quantities in the planned area. Thus, each objective function is expressed as the sum of two terms: the actual path length, time elapsed, or energy use of the planned section, and the smallest quantity of interest to travel from the last path segment's end point to the final destination. An example of this is seen in Figure 2, where the planned portion (yellow/blue) and the quantity of interest to travel to the final destination (black) is shown.

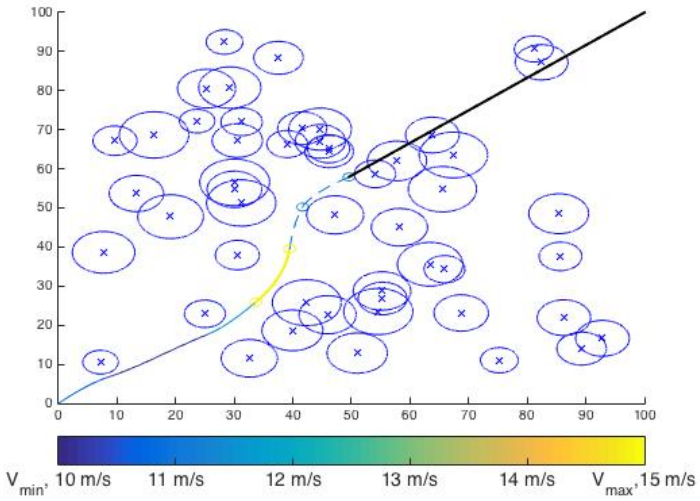


Figure 2: An illustration of how the objective function which minimizes path length is calculated. The sum of the length of the UAV's planned path (yellow/blue) and the distance from the final control point to the final destination (black) is minimized.

1. *Decrease Path Length*

The objective function used to minimize total path length needed to reward paths which brought the UAV closer to its final destination, while minimizing the planned path length. Thus, we defined F_{length}, the

objective function used to minimize total path length, as the sum of two terms: the length of the planned paths and the distance from the UAV's final destination to the last segment's end control point P_2. When m path segments are planned this function is expressed as:

$$F_{length} = \sum_{i=1}^{m} \ell_i + \ell_f \tag{3}$$

where ℓ_i is the length of the ith path and ℓ_f is the distance from the UAV's final destination to the last segment's end control point, which is defined as:

$$\ell_f = \sqrt{(x_f - x_{f,m})^2 + (y_f - y_{f,m})^2} \tag{4}$$

Note that $(x, y)_{f,m}$ are the coordinates of the last segment's end control point, P_2. It is difficult to analytically compute the length of a Bézier curve, so we estimated this length through numerical integration.

2. Decrease Time Elapsed

The objective function used to decrease the time elapsed traveling the path, F_{time}, was similarly defined. The function needed to reward paths which quickly brought the UAV toward its final destination. This is achieved by defining F_{time} as the sum of two terms: the time elapsed when the UAV traveled each planned path and the time elapsed if the UAV traveled at its maximum speed directly from the last planned segment's end control point, P_2, to the final destination. The time elapsed when the UAV travels a path segment is Δ_{time}, so the first term is simply $m\Delta_{time}$. Note that we used $\Delta_{time} = 1$ s. The time elapsed if the UAV travels at its maximum speed from the last planned segment's end control point is defined as ℓ_f / V_{max}, so F_{time} is

$$F_{time} = m\Delta_{time} + \frac{\ell_f}{V_{max}} \tag{5}$$

3. Decrease Energy Use

To define F_{energy}, the objective function which decreases energy use over the planned path, we determined the required energy of the UAV, which is defined as:

$$E_{req} = \frac{D\ell_{est}}{\eta_{overall}} \tag{6}$$

where D is the UAV drag and $\eta_{overall}$ is the propulsive efficiency. Thus, these quantities needed to be determined. We estimated the UAV drag using Carson's approach,[23] which takes into account flight conditions such as air density, parameters such as span, weight, and parasite area, and the UAV's velocity. Because all terms except the UAV velocity are constant, we modeled the drag as a function of the UAV's speed. This is seen in Figure 3.

The overall propulsion efficiency was also modeled as a function of V. For simplicity, in lieu of using 1st principles or experimental data to derive such a relationship we approximated this relationship as the function $\eta(V)$:

$$\eta(V) = \begin{cases} V \leq 10, \ 0.06V \\ 10 < V < 20, \ -0.0024V^3 + 0.084V^2 - 0.9V + 3.6 \\ V \geq 20, \ 0.00001 \end{cases}$$

A plot of this is seen in Figure 4, which is an approximation of the curves presented by Spakovsky.[24]

The minimum value of the quotient of the drag and efficiency D/η was set to $(D/\eta)_{opt}$. This value, multiplied by ℓ_f, is the minimum energy required by the UAV to reach its final destination from its current position. Using Equations 5 and the value $(\frac{D}{\eta})_{opt}$, we defined F_{energy}, the objective function which minimizes energy use:

$$F_{energy} = \sum_{i=1}^{m} \left[\int_0^{\ell_i} \frac{D_i}{\eta_i} dx \right] + \ell_f \left(\frac{D}{\eta} \right)_{opt} \tag{7}$$

which is solved through numerical integration.

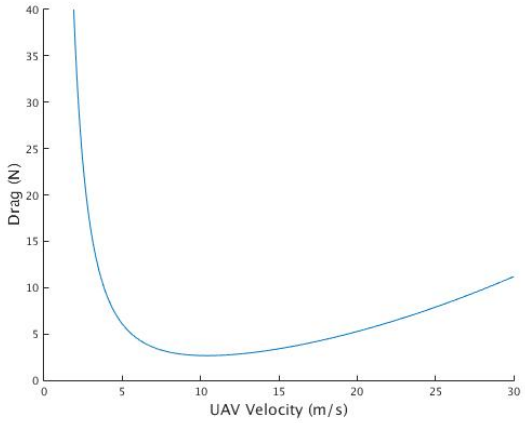


Figure 3: Total drag acting on UAV during flight.

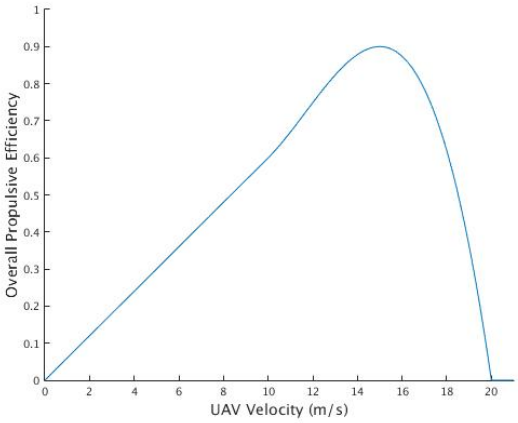


Figure 4: Propulsive efficiency of UAV.

D. Constraints

1. Obstacle Constraints

Constraints were used to ensure that the UAV avoided collision with static or dynamic obstacles. The distance from each obstacle (including its uncertainty bound) to the initial position of the UAV during each interval was calculated. If this distance was greater than $m\ell_{max}$ (the maximum distance the path could be planned in an interval), the obstacle was assumed to be outside the UAV's field of view and was neglected. Each segment was then discretized and constrained so that each point along the curve met the obstacle avoidance requirement for obstacles within the UAV's field of view. Each segment was sampled at $t = 0, 0.05, 0.1, \ldots, 0.95, 1$. These constraints then took the form

$$r_j \leq \sqrt{(x(t)_i - x_j)^2 + (y(t)_i - y_j)^2} \tag{8}$$

where r_j and (x_j, y_j) are the radius and position of the j^{th} obstacle, respectively. Similar constraints were computed for dynamic obstacles. However, instead of a fixed (x_j, y_j), it was predicted where the dynamic obstacle would be based on its known trajectory to avoid it at that specific point in time.

2. Flight Constraints

Common aerodynamic flight constraints were also added. Fixed-wing UAVs are limited by their stall and maximum speed. Each path segment planned corresponded to some time elapsed while flying, so a maximum path segment length correlated to the UAV's maximum speed and a minimum path segment length correlated to the UAV's minimum speed. For our simulations, we used $\ell_{min} = 10$ m to represent V_{min} and $\ell_{max} = 15$ m to represent V_{max}. Since $\Delta_{time} = 1$ s, $V_{min} = 10$ m/s and $V_{max} = 15$ m/s. Thus,

$$\ell_{min} \leq \ell_{est,i} \leq \ell_{max} \tag{9}$$

The minimum curvature of the Bézier curve (k) was constrained so that it was greater than the minimum turn radius r_{turn} of the UAV, where

$$k \geq r_{turn} \tag{10}$$

Note that k was calculated in the same manner as done by Oz.[9] Note that we set $r_{turn} = 5$ m. We also constrained the first derivatives of the paths so that the derivative at the last point of one segment was equal to the derivative of the first point of the following segment. This is described as

$$B'(t = 1)_{i-1} = B'(t = 0)_i \tag{11}$$

which better represents realistic flight preventing instantaneous direction change.

E. Removal of Infeasible Paths & Multi-start Approach

1. *Removal of Infeasible Path Segments*

On occasion, one or more of the solutions for the planned path segment did not converge, and resulted in an infeasible path segment. If a solution had converged to an infeasible point, such as violating an obstacle avoidance constraint, the segment was removed. If it used a single-start approach, the solver quit, but if a multi-start approach was used, then just that solution would be removed. An example of this can be seen in Figures 5 and 6. In Figure 5, infeasible path segment solutions were not removed and were thus chosen. Once these solutions were removed, a viable path that avoids the obstacles was planned, as seen in Figure 6.

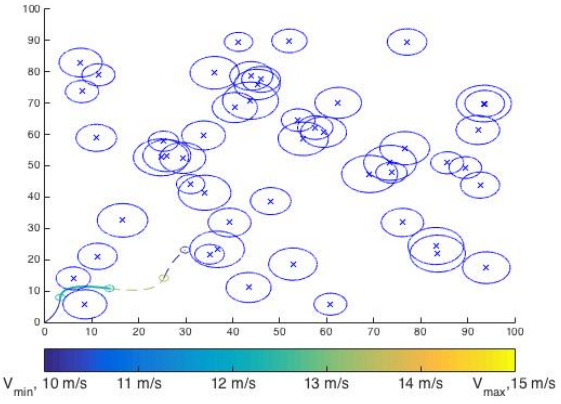


Figure 5: A path with infeasible segments not removed, resulting in obstacle collision

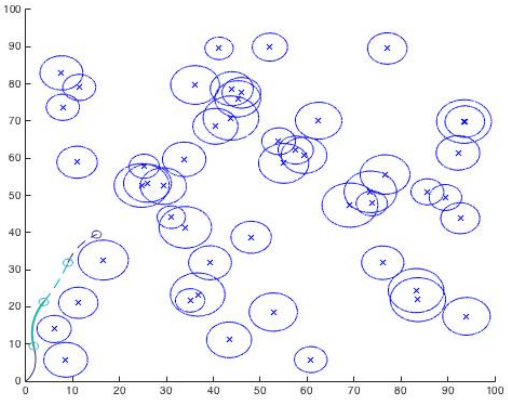


Figure 6: A path with infeasible segments removed, resulting in a viable path

2. *Multi-start Approach*

We used two methods to generate initial guesses. Generally, three planning segments were used in the optimization problem. At the beginning of each planning step, the initial guess of the first two segments was set to the second and third segments found from the previous step, or $S_{1,i} = S_{2,i-1}$ and $S_{2,i} = S_{3,i-1}$. The third segment was then defined as a path of length $0.75\ell_{max}$, oriented toward (x_f, y_f). This method reduced the time to convergence because the algorithm did not have to "redo the work" it performed in the previous planning step.

Two more initial guesses were supplied in this approach. The two guesses were initially oriented toward the final destination and then gradually curved to either a $+x$ or $+y$ direction respectively. The initial guess which curved to a $+x$ direction was set by

$$(x, y)_{guess} = (x_{0,i} + i\ell_{max},\ y_{0,i} + \frac{(i - 2i^2)}{m}\ell_{max}) \tag{12}$$

where $(x_{0,i}, y_{0,i}) = P_{0,i}$ and i is the current path segment. A similar guess was implemented for the $+y$ curve. The algorithm then ran for each initial guess, and then used the best solution from these three iterations. Despite increasing computation time, this greatly increased the method's robustness, and all results were found using a multi-start approach.

III. Results

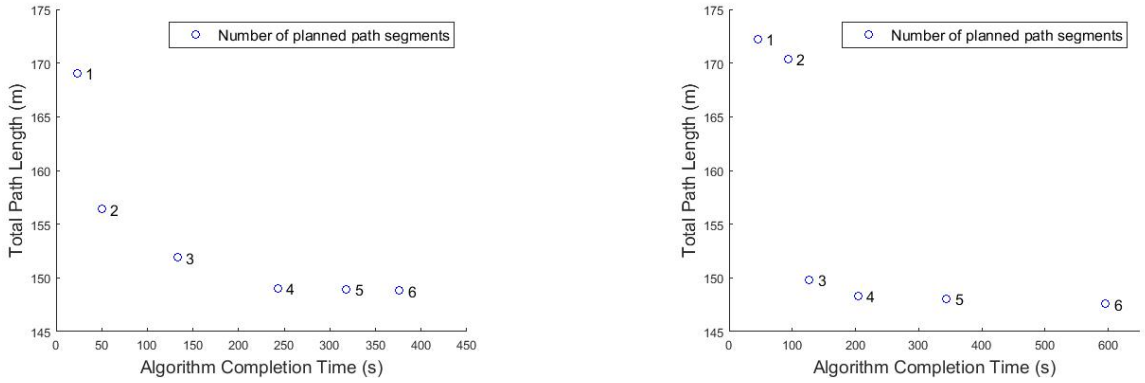
We tested our approach on several different obstacle fields, which contained static or dynamic obstacles. Paths planned using the three objective functions defined above were also compared. For the subsequent results, three path segments were planned in each iteration. This is justified in Section A, where comparisons of flight paths using varying numbers of planned path segments are given. Section B shows the planning progression through an obstacle field comprised of only static obstacles, as well as the planning progression

through a flight field of three dynamic obstacles. Comparisons of flight paths planned using different objective functions are seen in Section C. Finally, we compared the planned paths using this receding horizon approach to optimal paths, where the entire obstacle field is known.

A. Number of Planned Path Segments Comparison

We first considered how the number of planned path segments would affect the overall path. This comparison was used to determine how far the optimal "horizon" is for the receding horizon approach. We varied the number of planned path segments from one planned path segment (no look-ahead segments), to six planned path segments (five look-ahead segments). This was done in two different static obstacle fields.

Using these obstacle fields, we compared the overall path length versus the computation time, which is seen in Figures 7a and 7b. From these figures, when more path segments are planned out the resulting path's total length is shorter but more computationally expensive. By inspection, using more than four path segments does not decrease the path length significantly but does contribute to a longer computation time. In addition, only planning one or two look-ahead segments often results in poor paths.



(a) 2nd obstacle field

(b) 1st obstacle field

Figure 7: Number of planned path segments comparison through static obstacle fields. As additional paths segments are planned each iteration, the computation time increases.

In regard to these and other static obstacle fields, using three or four path segments led to the best results. These paths were not too computationally expensive, while still planning a near-optimal path. Thus, all subsequent results plan three segments.

B. Obstacle Avoidance

Figure 8 shows the path planned while minimizing path length. In this obstacle field, 52 static obstacles were placed of size $3.5 \leq r_j \leq 6.5$. At each iteration, the algorithm used a receding horizon approach to plan each path segment, and the segment is traversed as the UAV nears its final destination. As can be seen, the UAV is able to successfully avoid static obstacles and complete its mission. Videos of the path planning iterations are available online[a].

A situation where dynamic obstacles are in the UAV's flight zone was then constructed, as shown in Figure 9. F_{length} is again used as the objective function. The obstacles had radii of $r_d = 7$ m, initial positions of $(75, 25)$ m, $(95, 25)$ m, and $(75, 5)$ m, and a known trajectory of $\vec{v} = (-6.25, -6.25)$ m/s. A single static obstacle is also present, but doesn't affect the UAV's path. In each frame of Figure 9, the dark, bold red circles represent the dynamic obstacles' positions at the beginning of the UAV path segment, and the thin dashed, dark red circle represent the obstacles' positions when the UAV has traversed the planned segment.

As can be seen from Figure 9, the UAV is able to weave around the dynamic obstacles and avoid collision while minimizing the total path length.

[a]https://youtu.be/YzZnfn4yCLs

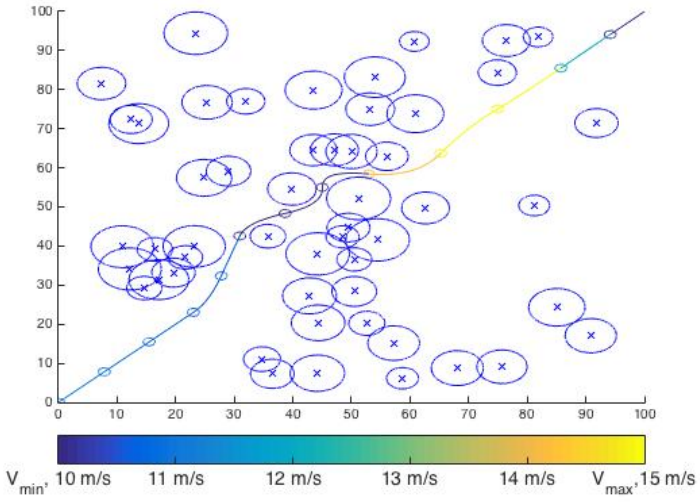


Figure 8: An example of the path planning approach. The UAV's path (yellow/blue) successfully avoids the static obstacles (blue).

C. Path Comparisons

We constructed a situation where two dynamic obstacles are in the UAV's flight zone, shown in Figure 10. The first objects' radii was $r_{d,1} = 10$ m, initial position $(75, 25)$ m, and trajectory $\vec{v}_1 = (-4, 4)$ m/s. The second object's radii was $r_{d,2} = 10$ m, initial position $(25, 75)$ m, and trajectory $\vec{v}_2 = (4, -4)$ m/s. In each frame of Figure 10, the bold red circles represent the dynamic obstacles' positions at the beginning of the UAV path segment, and the thin dashed red circles represent the obstacles' positions when the UAV has traversed the planned segment. Because of the algorithm's setup, a single static obstacle is also present, but doesn't affect the UAV's path. The objective in Figure 10 was to decrease the length of the UAV path. As can be seen from the figure, the UAV avoids collision with the dynamic obstacles.

The objective function was then changed to F_{time} to minimize the time elapsed for the UAV to reach its final destination, where the evolution of the planned path is shown in Figure 11. As can be seen, the path directs the UAV around the two dynamic obstacles. This is significantly different than the path planned to minimize distance. F_{energy} was then set as the objective function. It's path was in the same trajectory as the minimize path length flight, but took less time. A comparison of this path is seen in Table 1 to the other two planned paths.

Table 1: Path comparison over dynamic obstacle field

	Time elapsed (s)	Path Length (m)	Energy use (J)
Minimize Path Length	14.0	141.4214	634.9598
Minimize Time	10.0	148.1410	554.2568
Minimize Energy Use	11.0	141.4214	515.5476

A comparison of paths planned by the three objective functions was also done over a static obstacle field. This can be seen in Figures 12a, 12b, and 12c. As can be seen from these figures and Table 2, the path which minimizes energy use required less energy than the paths which minimizes time and path length. The path which minimizes time required the UAV to travel at or near its maximum speed, and the path which minimizes path length travels near the UAV's minimum speed. As can be seen from Figures 3 and 4, traveling at these speeds results in increased drag and a low propulsive efficiency, which increases the energy use. Conversely, the path which decreases energy use planned the UAV to travel at a more energy efficient velocity. This speed is lower than the maximum speed, resulting in more time elapsed. In addition, the path which decreases minimizes length also results in more time elapsed.

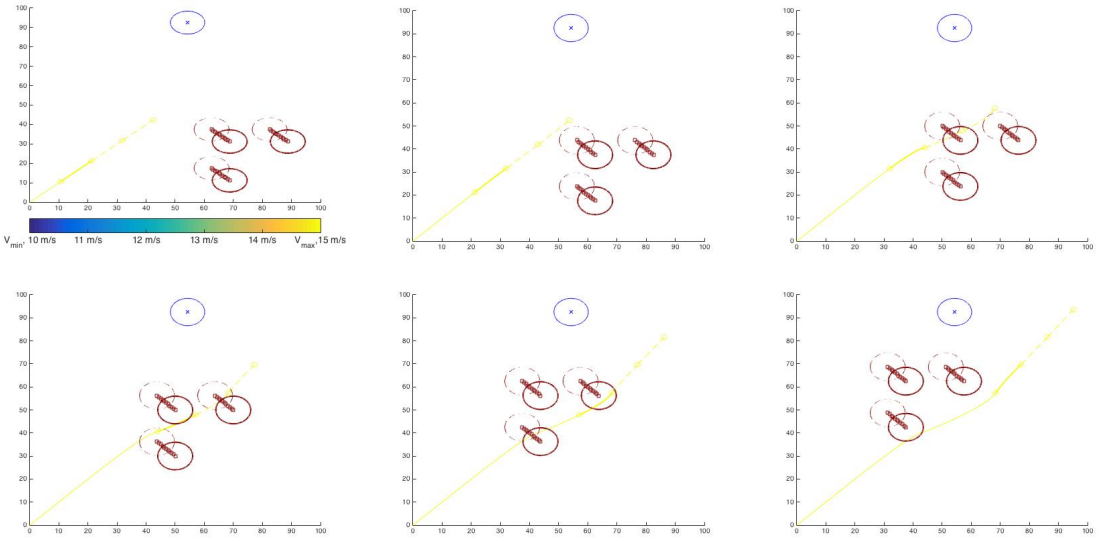


Figure 9: An example of dynamic obstacle avoidance while minimizing path length. The UAV (yellow/blue) is able to avoid the dynamic obstacles (dark red).

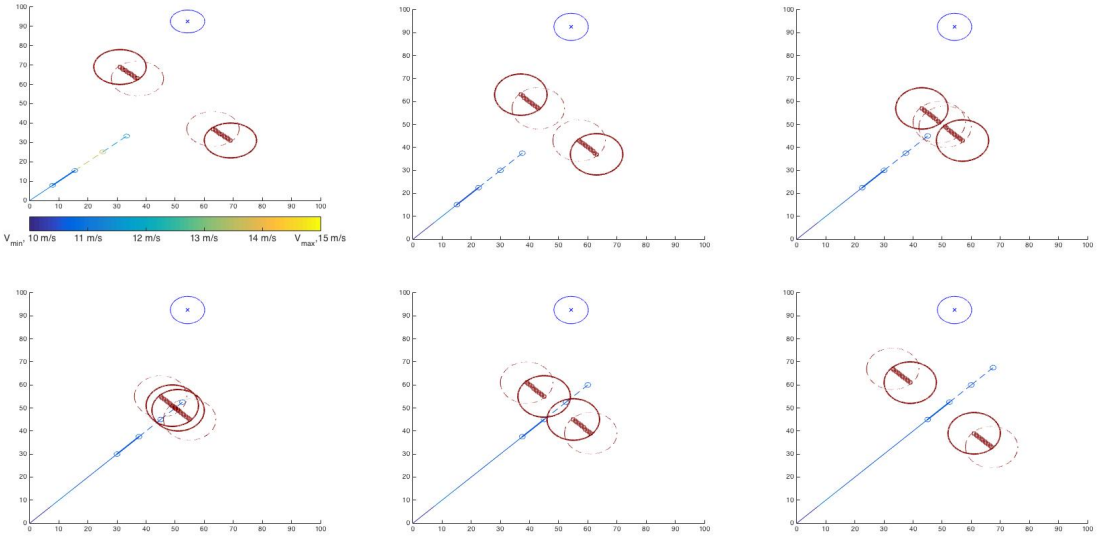


Figure 10: An example of dynamic obstacle avoidance while minimizing path length. The UAV (yellow/blue) is able to avoid the dynamic obstacles (dark red). Note the difference between this planned path and that in the following figure.

Table 2: Path comparison over static obstacle field

	Energy Use (J)	Time elapsed (s)	Path Length (m)
Minimize Energy Use	558.1766	12.0	150.5044
Minimize Time	578.7844	11.0	152.1642
Minimize Path Length	657.7634	14.0	146.3228

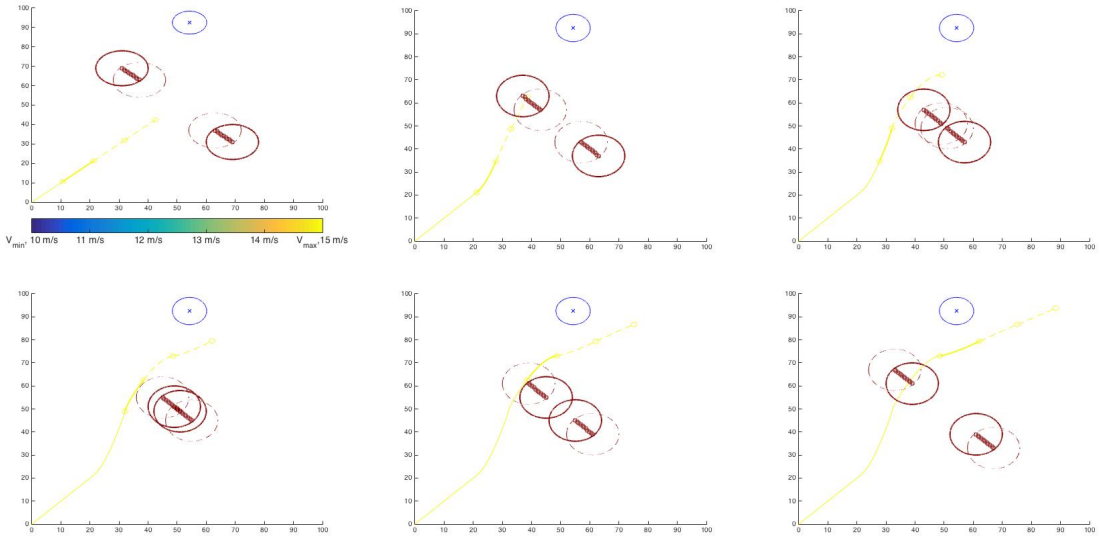
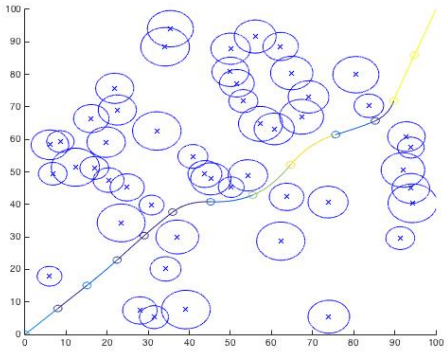
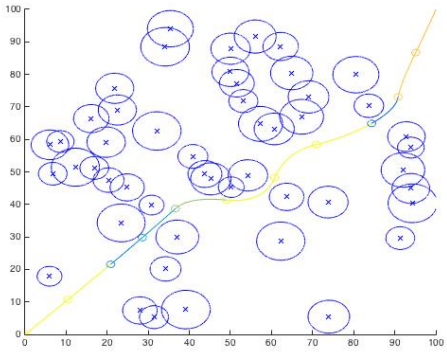


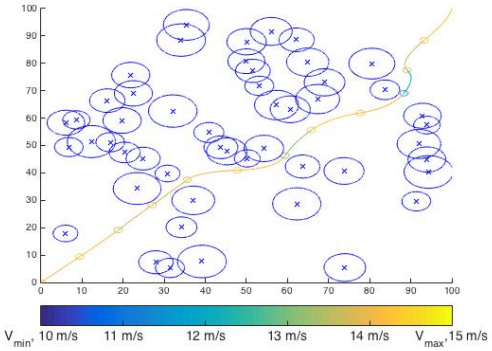
Figure 11: An example of dynamic obstacle avoidance while minimizing time. The UAV (yellow/blue) is able to avoid the dynamic obstacles (dark red). Note the difference between this planned path and that in the previous figure.



(a) Minimum path length



(b) Minimum time

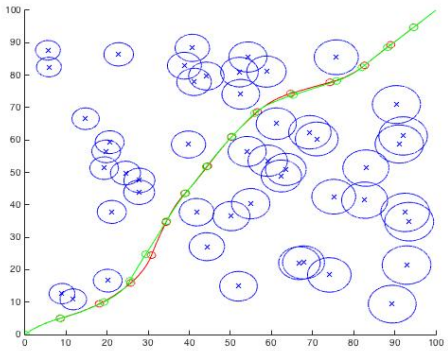


(c) Minimum energy use

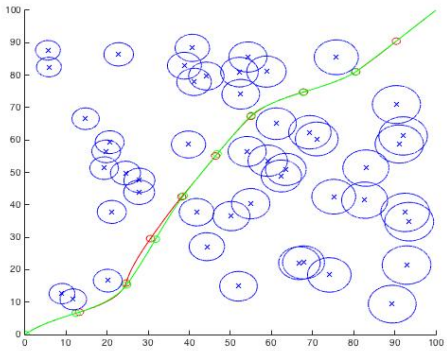
Figure 12: Examples of static obstacle avoidance while optimizing the planned path. The UAV (yellow/blue) successfully navigates through the obstacles (blue).

D. Comparison to Full-Knowledge Optimal Paths

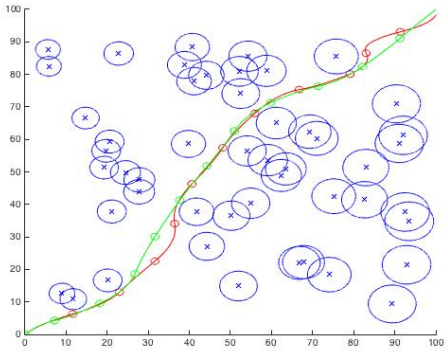
For ten different static obstacle fields, we compared the objective function's value of the planned paths using our described approach to a full-knowledge optimal solution. We first compared a path that was planned using F_{length} as its objective function to a true minimum length path. An example of this is shown in Figure 13a, where the planned path is green and the optimal path is red. On average, the full-knowledge optimal path was 3.3% shorter than the limited-knowledge optimal path. Similar results were found for a path that was planned using F_{time} as its objective function to a true minimum time path. An example of this is seen in Figure 13b, and on average, the optimal time paths were completed in 0.9% less time. A comparison was also done for a path that was planned using F_{energy} as its objective function to a true minimum energy use path. An example is shown in Figure 13c, and on average, the optimal energy use paths used 2.6% less energy than the planned paths.



(a) Minimum path length



(b) Minimum time



(c) Minimum energy use

Figure 13: Comparisons that shows a planned path (red) using our approach to a full-knowledge optimal path (green) while avoiding static obstacles (blue).

IV. Conclusions / Looking Ahead

We tested our approach in a myriad of different obstacle fields, which contained static and/or dynamic obstacles. Comparisons of flight paths planned using different objective functions were also included, as well as comparisons of flight paths using varying numbers of planned path segments. As discussed in the previous section, this method was able to successfully detect and avoid obstacles, while planning a near optimal path. The algorithm planned paths that avoided obstacle collision in flight fields comprised of static or dynamic obstacles. Comparisons were also drawn between paths that were planned using different objective functions, as well as paths that used different numbers of look-ahead segments.

There are a number of straightforward extensions to this path planning method. These include considering non-circular obstacles and extending to 3D space. Though implementing obstacles of different shapes would not be difficult, it would require redefinition of the constraints. The way the optimization is configured, adding more complex shapes would present additional computational costs. It may also may be just as efficient to model complex shapes with overlapping circles. In addition, the method here only considers trajectories in a 2D space, but the trajectories could be extended to 3D space by extending the 2D Bezier splines into 3D space.[4]

It would also be beneficial to determine how the receding horizon approach could be more robustly implemented. If the optimizer were to get truly stuck, perhaps in a wide, deep concavity, some method of picking a new route would be necessary. One possibility is to keep a map of the space and update regions as visited as the path goes through them. If the planner gets stuck, it could then pick a favorable not-previously-visited region of the map, backtrack to a branch point, and proceed into the new region.

One challenge for this approach is that we assume trajectories of dynamic obstacles in the flight domain are known within some uncertainty bounds. Though this information can be communicated for UAVs connected to an unmanned aerial system (UAS), aircraft with unknown trajectories and other dynamic obstacles (such as wildlife) would require larger uncertainty bounds.

References

[1]S. Bortoff, *Path planning for UAVs* In *Proc. of the American Control Conference*, pp. 364-368, Chicago, IL, 2000.

[2]S. LaValle, *Rapidly-exploring random trees: a new tool for path planning* 1998.

[3]E. Mansury, A. Nikookar, and M. Salehi, *Differential evolution optimization of ferguson splines for soccer robot path planning.* In *Innovations in Intelligent Machines*, S. C. Javaan, C. J. Lakhmi, M. Akiko, and S.-I. Mika, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 77111.

[4]I. Hasircioglu, H. Topcuoglu and M. Ermis, *3-D path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms.* In 10*th Annual Conference of Genetic Evolutionary Computation*, 2008.

[5]I. Nikolos, E. Zografos, and A. Brintaki, *Heuristic methods for randomized path planning in potential fields.* In Ali Movaghar, Mansour Jamzad, and Hossein Asadi, editors, *Artificial Intelligence and Signal Processing*, volume 427 of *Communications in Computer and Information Science*, pp. 311-319. Springer International Publishing, 2014.

[6]J. Baker, *Reducing Bias and Inefficiency in the Selection Algorithm.* In *Proceedings of the Second International Conference on Genetic Algorithms and Their Application*, pp. 14-21. L. Erlbaum Associates Inc., 1987.

[7]J. Tisdale, Z. Kim, and J. Hedrik *Autonomous UAV path planning and estimation.* In *IEEE Robot. Autom. Mag.*, vol. 16, no.2, pp. 35-42, Jun. 2009.

[8]S. Caselli, M. Reggiani, and R. Rocchi, *UAV path planning using evolutionary algorithms.* In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 426-431, Banff, Alberta, Canada, 2001.

[9]I. Oz, H. Topcuoglu, and M. Ermis, *A meta-heuristic based three-dimensional path planning environment for unmanned aerial vehicles* In *Simulation : Transactions of the Society for Modeling and Simulation International*, 2012.

[10]K. Kulling, *Optimal and Receding-Horizon Path Planning Algorithms for Communications Relay Vehicles in Complex Environments.* Massachusetts Institute of Technology, 2007.

[11]D. Dathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi, *An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments.* In 21*st digital avionics systems conference* (*Vol.*2).

[12]T. Fraichard and A. Schuer, *From Reeds and Shepp's to continuous-curvature paths.* IEEE Transactions on Robotics, pp. 1025-1035, IEEE, 2004.

[13]X. Z. Gao, Z. X. Hou, X. F. Zhu, J. T. Zhang, and X. Q. Chen, *The shortest path planning for manoeuvres of UAV.* Acta Polytechnical Hungarica, Vol. 10, No. 1, 221-239, 2013.

[14]M. Saska, M. Macas, L. Preucil and L. Lhotska, *Robot path planning using particle swarm optimization of ferguson splines.* IEEE Conference on Emerging Technologies and Factory Automation, 2006.

[15]E. Mansury, *Artificial bee colony optimization of Ferguson splines for soccer robot path planning.* In *Proceeding of the* 2013 *RSI International Conference on Robotics and Mechatronics*, volume 1, pp. 5-89, 2013.

[16]M. Lizarraga and G. Elkaim, *Spatially deconflicted path generation for multiple UAVs in a bounded airspace.* ION/IEEE Position, Location, and Navigation Symposium, ION/IEEE PLANS 2008, Monterey, CA, May 5-8, 2008.

[17]S. Upadhyay and A. Ratnoo, *Smooth trajectory planning for MAVs with airspace restrictions.* AIAA Guidance, Navigation, and Control Conference, 4-8 January 2016.

[18]J. Bellingham, M. Tillerson, M. Alighanbari and J. How, *Cooperative path planning for multiple UAVs in dynamic and uncertain environments.* In *Proceedings of the* 41*st IEEE conference on decision and control*, pp. 28162822, 2002.

[19]X. Peng and D. Xu, *Intelligent online path planning for UAVs in adversarial environments.* Northwestern Polytechnical University, 2012.

[20]W. Al-Sabban, L. Gonzalez, R. Smith, and G. Wyeth, *Wind-energy based path planning for electric unmanned aerial vehicles using Markov decision processes.* Queensland University of Technology.

[21]G. Nachmani, *Minimum-Energy flight paths for UAVs using mesoscale wind forecasts and approximate dynamic programming.* Naval Post Graduate School, 2007.

[22]K. Joy, *Quadratic Bézier Curves.* University of California, Davis; 2000.

[23]B. Carson, *Fuel Efficiency of Small Aircraft.* U.S. Naval Academy, Annapolis, 1980.

[24]Z. Spakovsky, *Thermodynamics and Propulsion, 11.7: Performance of Propellors.* Massachusetts Institute of Technology, 2009.