2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Coordinated commencement of pre-planned routes for fixed-wing UAS starting from arbitrary locations - a near real-time solution*

James Keller[†], Dinesh Thakur[†], Vladimir Dobrokhodov[§], Kevin Jones[§], Maxim Likhachev[‡], Jean Gallier[†], Isaac Kaminer[§] and Vijay Kumar[†]
[†] GRASP Laboratory, University of Pennsylvania, Philadelphia PA
[‡] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA
[§] Department of Mechanical and Aerospace Engineering, Naval Postgraduate School, Monterey, CA

Abstract— **Coordinated time optimal path planning and trajectory management algorithms for air vehicles depend on precise simultaneous mission commencement by all agents. Ground-based and rotary-wing aerial vehicles can be staged at their mission initial conditions until a mission is commenced but fixed-wing aerial vehicles, which must maintain a minimum forward airspeed at all times, can only be positioned approximately. A computationally simple algorithm for these vehicles that determines simultaneous arrival paths from arbitrary starting points is presented. The algorithm is based on planar B-spline curves so that fully defined feasible trajectories can be quickly determined, compactly encoded, and precisely executed.**

I. INTRODUCTION

The complexity of planning trajectories for coordinated time-critical Unmanned Aerial System (UAS) operations typically results in computational times on the order of multiple seconds or longer depending on the processor used, the dimensionality of the given problem and the rigor with which physical constraints are included [1]. When working with fixed-wing agents, which must maintain a minimum forward speed, planning algorithms must, therefore, assume starting locations since the real-time solution based on actual locations is not viable. To address this, vehicles are typically staged in nearby loitering flight plans until the mission execution command is issued. Specific applications may call for predetermined holding patterns to account for obstacles or ensure path deconfliction. However, the performance of a typical autopilot does not permit vehicles starting at somewhat arbitrary locations and flight path vectors to arrive in concert at the initial points of their flight plans. In some cases, adequate performance can be obtained by buffering the planned mission with an initial segment for each agent to account for this variability. If inter-vehicle communications permit vehicle locations to be shared, then the autopilots can be configured to coordinate approximate synchronized completion of all buffered segments. Since the airspeed range available to regulate coordination is limited and diminishes as ambient temperature and altitude are increased, performance guarantees for such strategies do not exist. The variability of winds aloft further complicates the ground-referenced coordination problem to the extent that any available airspeed variability is best allocated to maintaining coordination not establishing it. Consequently, it is useful to have a near real-time planner that is capable of reacting to instantaneous vehicle location and mission start points and their respective flight path parameters to determine a precise time of arrival for each that can be tailored to ensure simultaneity of arrival for the team. Continuity of curvature and constraints on its magnitude and rate of change are also critical components of feasible paths because the response time to turning commands for the small UAS vehicles typically used in this application can be significant. Strict time optimality is not critical but candidate paths should be reasonably efficient because mission duration is also a priority. The critical requirements for a practical simultaneous arrival algorithm are that paths must be feasible, must be computed in near real-time, and be free of potential collisions. Collision deconfliction has two solutions:

1) space deconfliction of paths
2) time deconfliction of velocity profiles

Optimization algorithms can be formulated to address the constrained problem as described in [2], [3], or [4] but these approaches are computationally expensive and not likely to meet the near real-time requirement on an embedded processor. A solution to the fully constrained problem may, however, be critical in cases where collision deconfliction requirements become complex. An approach tailored for embedded processing is developed by Nelson et al [5] in which C^1 continuity is achieved. This paper describes an algorithm that can be rapidly executed even with the limited onboard computing resources and yield C^2 continuity with planned missions. In our case we address collision deconfliction by separation of initial holding patterns to minimize problem complexity prior to the mission start. We limit constraints to feasibility of path considerations to further minimize complexity.

An example of coordinated flight requiring such considerations is presented on Figure 1, which shows the flight plans for two vehicles so they can simultaneously observe a road segment from opposite sides. In order for this mission to be successful, each vehicle must maintain precise coordination

*This work was supported by the U.S. Office of Naval Research grants N00014-09-1-1051 and N00014-09-1-103

with its teammate as the road is traversed so that there is a single sensor location for both. Complex missions may require coordination of more than two vehicles.

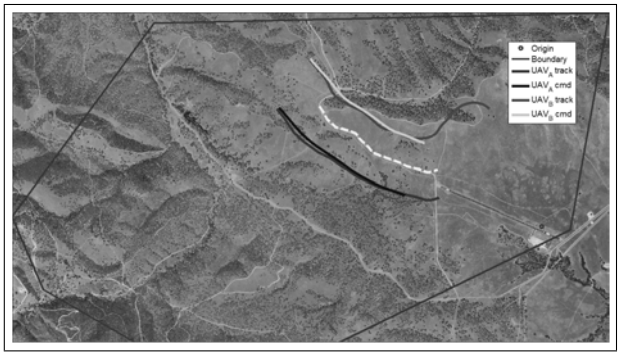

Fig. 1. Example of a coordinated road search

II. Development of Simultaneous Arrival Plans: Top Level Requirements

Of the many approaches that can be taken toward simplification of the development of a curve between the instantaneous states of a vehicle and the initial segment of its planned route, complexity can be reduced by limiting the problem to planar, constant altitude, curves. Constant airspeed planning further reduces complexity and reserves that degree of freedom for real-time trajectory corrections. With these considerations, Dubins' paths[1] can be quickly computed but they only support C^1 continuity and are consequently not considered to be feasible. Moreover, they are not suited for subsequent operations to tailor path duration. The effects of feasibility constraints on a solution become more significant as initial and final points are brought into close proximity[6]. Therefore, a second step to reduce the complexity of the problem is to sufficiently separate the points. In the simple case of circular orbits, this relates to the centers about which the vehicles are staged and the planned mission start points. For more complex holding patterns, the closest point of the pattern with respect to the planned mission start can be used to establish the center of the tightest feasible turn towards the mission. In this manner a relatively simple path can be determined which can be adjusted to meet feasibility constraints with respect to curvature. A Dubins-like geometric construction can be used to determine a working boundary condition separation by replacing the actual minimum turning radius with one that has been expanded to account for the finite period of time required to generate a maximum turning rate; see Scheuer and Fraichard [7] or Scheuer and Laugier [8] for their near-optimal path construction using clothoid curves, which is illustrated on Figure 2. This process can be applied when clothoid segments are used to approximate curvature variation along a path so that continuous curvature paths can be constructed, see [9] or [10] for applications. Scheuer's extended radius construction concept can also be generalized beyond clothoid curves to provide a better approximation of the actual path taken by a fixed-wing aircraft from straight and level flight to turning flight at minimum radius and back to straight flight. However, when aerodynamic and limited control authority constraints are imposed, the turn rate time history is no longer symmetric, so the construction is *not* exact. The underlying equations of motion we use for turning segments, may be found in texts such as Seckel [11] or Stengel [12]. Denoting the heading angle as ψ, the gravitational constant as g, roll rate as p, roll moment normalized by inertia as L, the angle of bank as ϕ, the roll control as $\delta_{aileron}$ and speed as V, the turning equations of motion can be approximated as follows:

$$\begin{aligned} \dot{p} &= \frac{\partial L}{\partial p} p + \frac{\partial L}{\partial \delta_{aileron}} \delta_{aileron} \\ \dot{\phi} &\cong p \\ \dot{\psi} &= \frac{g \tan \phi}{V} \end{aligned} \tag{1}$$

These equations have been used to extend Scheuer's geometric construction as illustrated on Figure 3 for the maneuver performance of our vehicles. The right panel illustrates the aforementioned approximation of this approach, which only slightly compounds the sub optimality of the path. The path shape from maximum curvature to zero curvature can be precomputed based on the vehicles being used so that the construction can be applied to determine the best point from which to depart the loiter orbit. While path shapes could have been synthesized directly with Dubins' constructions, the extension lends itself to curvature-continuous paths when endpoints are sufficiently separated.

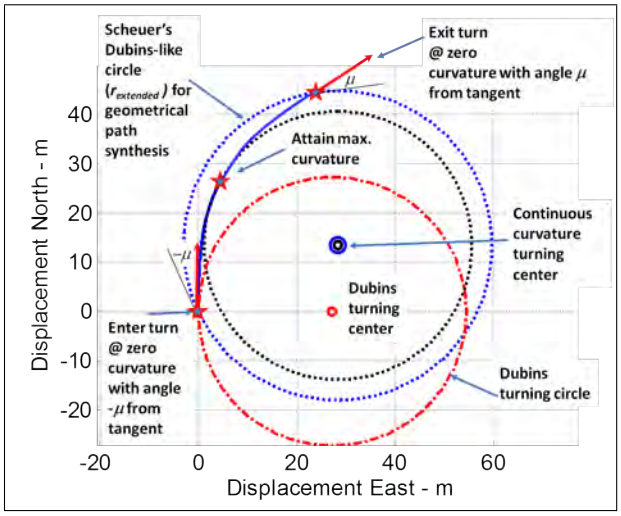

Fig. 2. Scheuer's Dubins-like path construction for continuous curvature paths (as derived for clothoid spirals)

A. Minimum proximity of IC orbits to mission plans

In order to derive a feasible path that can be determined with minimal computational complexity, we apply Scheuer's

[1]Dubins' paths are optimal with respect to path length and are constructed of curved segments at minimum radius and straight segments. Dubins showed that optimal paths have the form of $\langle curve - straight - curve \rangle$ or $\langle curve - curve - curve \rangle$, where curves can either be to the left or to the right. Dubins showed the optimal path is an element of the set {LSL, RSR, LSR, RSL, LLL, RRR}

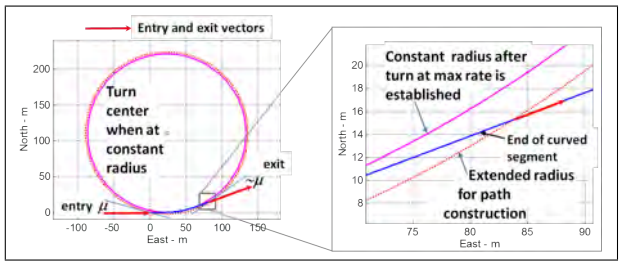

Fig. 3. Scheuer's Dubins-like path construction for continuous curvature paths extended to equations of motion 1.

construction extended to the simplified equations of motion, (1), to identify a minimum spacing. We use the constraints that arise from a path with inflection (LSR or RSL) since these cease to exist when endpoints are too close[2]. In the case of an arbitrary holding pattern, instantaneous curvature can be used to locate a point on the Scheuer construction entry to a circular orbit and the affine nature of the construction can be used to align the path with current position and flight path vector. In this manner, any holding pattern can be mapped to an equivalent circular orbit. Without loss of generality we assess this construction for a clockwise starting orbit to a point with arbitrary direction (path tangent) and curvature (within the feasible peak used to size the orbit radius)[3]. Once bounds are set on the distance between IC orbit departure and mission starting points such that this construction is feasible, a feasible parametric polynomial curve between orbit departure and mission start can also be found to approximate it. The geometry of Scheuer's construction in this case is illustrated on Figure 4, note the additional segment that must be considered to transition from an arbitrary holding pattern to a turning pattern. We use this construction to derive the minimum spacing to execute a turn from a clockwise direction from a point **A** to a goal point **C**, with maximum curvature in the CCW direction, or **D**, with maximum curvature in the CW direction as candidate paths. The locations and orientations of goal points are then generalized so that the distance from orbit to mission start required to guarantee feasibility can be bounded for arbitrary parameterization. The feasibility of the construction can be easily verified by following the maneuver on which it is based. The critical intermediate segment follows the path from either **A** (if vehicle is currently on a zero curvature portion of an arbitrary holding pattern) or a point **A**′ if on a simple orbit to a point denoted as **B** with zero curvature. This is the closest location for which zero curvature can be achieved using a RSL maneuver to **B** from **A** (or **A**′). Note, in this case the straight segment would have zero length and only entail an inflection from R to L turning segments. From point **B** with orientation as drawn to a condition of arbitrary curvature turning CCW, the terminal point can then be located along the segment from **B** to **C**, since **C** is the earliest location where curvature can be maximal CCW from **B**. Similarly, for arbitrary terminal curvature turning CW from **B**, the terminal point would lie along the segment from **B** to **D**. Since these segments are constructed from an affine map, they can be rotated to accommodate any arbitrary direction. Orientation of the flight path vector at point **B** other than that shown in the figure would cause the turning center of the construction to rotate about point **B** as shown by the heavy dotted circle. The general case of feasible goal points lies somewhere inside the dashed circle that intersects points **C** and **D**. The furthest point from an IC orbit point at **A** is illustrated as point **E**. In this regard, as long as the spacing between the orbit departure point (depicted as **A**) and the mission start point is greater than the distance from **A** to **E**, then this construction will exist and, hence, the path will be feasible. Rather than compute this distance precisely for specific boundary conditions, it can be bounded as $\geq 5r_{augmented}$, which can easily be verified visually from Figure 4. Furthermore, since orbit departure location/direction must be considered arbitrary, orbit centers (selected by the user) must be further than $\geq 5r_{augmented}+r_{orbit}$ from the start of the planned mission. While this establishes a reasonable bound on feasible paths, it does not consider the consequence of lengthening a segment to tailor duration. Since we do not want to depend on loiter orbit synchronization, in the worst case, a path may have to be lengthened by the full circumference of the orbit to achieve team simultaneous arrival. We overestimate this spacing to $\geq 8r_{augmented}$ so that the derived parametric curves require minimal complexity to satisfy time of arrival and feasibility constraints. For our vehicles, the duration of this traverse is less than forty seconds. Closer proximity is feasible if more constraints are included but we work with relatively slack constraints so the subsequent determination of path extensions required to satisfy simultaneous arrival requirements and feasibility constraints is straightforward. We do not expressly use this geometry to construct paths but rather to establish bounds on the existence of feasible paths.

[2]In the event the opposing curve, $\langle curve - straight - curve\rangle$ (LSR or RSL), path types cease to exist, an arbitrary feasible path may involve a $\langle curve - curve - curve\rangle$ construction, which is unnecessarily complex for our application. Curves of the form RSR or LSL always exist. Hence, we use a constraint which permits inflection in the limit to set boundary conditions.

[3]The geometry of the construction for a counter-clockwise IC orbit is simply a mirror image. Since the terminal point has an arbitrary state in our example case (direction, curvature), there is no loss of generality.

III. Application of a Parametric Curve Framework

Parametric polynomial curves are a natural framework for this application. Complete trajectories can be encoded with a small set of control points. In [13] we documented how parametric curves can be directly integrated with autopilot speed and turning commands so that navigation commands can be rapidly computed based on an arbitrary vehicle location and the associated point on the trajectory curve. Computation requirements are minimal to determine points along a path as well as other parameters such as tangent, normal, and curvature. In this application, the approach we take to provide simultaneous arrival for all agents is to

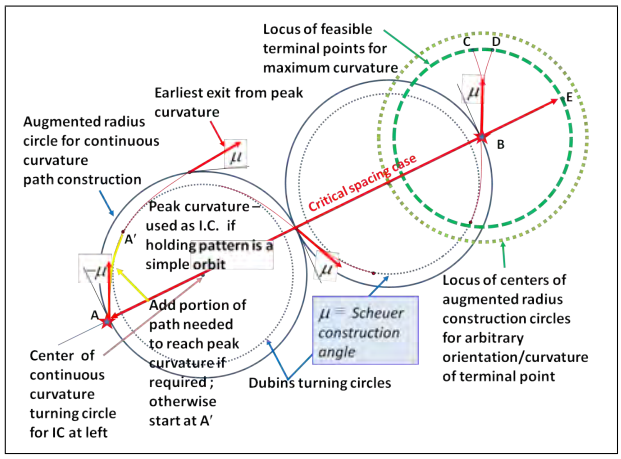

Fig. 4. Geometric construction of feasible path between IC orbit and mission starting point

first find paths solely based on boundary conditions using the fewest number of control points. These paths are then augmented to guarantee feasibility and finally additional control points are added to provide additional degrees of freedom so boundary conditions may be preserved whilst the interior portion of the shorter paths can be lengthened as required. The fewest number of control points that provides C^2 continuity at both ends with fixed boundary conditions is six[4]. Our algorithm can be configured to accept any number of agents but will be illustrated with two for simplicity.

A. Selection of an Appropriate Parametric Polynomial Basis

There are many polynomial bases with which parametric splines between our boundary conditions could be developed. Hermite and Legendre bases permit affine path constructions but do not guarantee curves will reside within the convex hull of their control points. Bernstein polynomials (Bezier curves and Pythagorean Hodographs) provide a convex hull property but do not permit local curve adjustment, which is useful for duration matching. Furthermore, Pythagorean Hodographs are too computationally complex for our near-real-time application but have been used in this context off-line [10]. Hence, we use B-splines to minimize computation and permit local curve shaping for duration matching. In [13], we showed that quartic splines provide sufficient degrees of freedom to approximate feasible trajectories for fixed-wing UAS, based on the equations of motion expressed in (1). We use clamped uniform spline basis functions[5] to ensure solutions are well conditioned numerically. The structure of the equation in this case is:

$$\boldsymbol{r}(\tau) = \sum_{m=0}^{5} N_{m,4}(\tau)\boldsymbol{p}_m, \text{ where } \tau \in [0,1)$$
$$\boldsymbol{p}_m \equiv \text{ set of 6 geometrically configured control points in } \mathbb{R}^2$$
$$N_{m,4}(\tau) \equiv \text{ set of 6 basis functions of degree 4} \tag{2}$$

Defining a curve solely on C^2 continuity boundary conditions establishes the following relations:

$$\begin{aligned}
\boldsymbol{r}(0) &= \sum_{m=0}^{5} N_{m,4}(0)\boldsymbol{p}_m \\
\boldsymbol{r'}(0) &= \sum_{m=0}^{5} N'_{m,4}(0)\boldsymbol{p}_m \\
\boldsymbol{r''}(0) &= \sum_{m=0}^{5} N''_{m,4}(0)\boldsymbol{p}_m \\
\boldsymbol{r''}(1) &= \sum_{m=0}^{5} N''_{m,4}(1)\boldsymbol{p}_m \\
\boldsymbol{r'}(1) &= \sum_{m=0}^{5} N'_{m,4}(1)\boldsymbol{p}_m \\
\boldsymbol{r}(1) &= \sum_{m=0}^{5} N_{m,4}(1)\boldsymbol{p}_m
\end{aligned} \tag{3}$$

Note, since we use clamped uniform B-splines, (3) can be expressed numerically (for $\boldsymbol{r}(\tau)$ and each element of $\boldsymbol{p}_m$ in $\mathbb{R}^2$), as follows:

$$\begin{bmatrix} \boldsymbol{r}(0) \\ \boldsymbol{r'}(0) \\ \boldsymbol{r''}(0) \\ \boldsymbol{r''}(1) \\ \boldsymbol{r'}(1) \\ \boldsymbol{r}(1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -4 & 4 & 0 & 0 & 0 & 0 \\ 12 & -18 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & -18 & 12 \\ 0 & 0 & 0 & 0 & -4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \boldsymbol{p}_3 \\ \boldsymbol{p}_4 \\ \boldsymbol{p}_5 \end{bmatrix} \tag{4}$$

The block diagonal structure of (4) permits solution through two decoupled equations:

$$\begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \frac{1}{4} & 0 \\ 1 & \frac{3}{4} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}(0) \\ \boldsymbol{r'}(0) \\ \boldsymbol{r''}(0) \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} \boldsymbol{p}_3 \\ \boldsymbol{p}_4 \\ \boldsymbol{p}_5 \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & -\frac{3}{4} & 1 \\ 0 & -\frac{1}{4} & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{r''}(1) \\ \boldsymbol{r'}(1) \\ \boldsymbol{r}(1) \end{bmatrix} \tag{6}$$

B. Determination of candidate feasible paths based on boundary conditions

Note, based on (6), when the direction of the planned route is approximately in line with the direction from the holding pattern/orbit, the last three control points of a connecting spline are determined solely by the start of the planned mission. When the direction of the planned path is orthogonal to the approach direction or opposes it, an additional Scheuer graphic construction, introduced in Figure

[4]Six control points permit location, first and second derivatives of the path to be matched at both ends but provides limited further independent degrees of freedom.

[5]nth order clamped B-splines use $n+1$ repeated knots at starting and ending parameters. In addition, to support a curve with m control points, the associated knot vector must be $m+n+1$ in length.

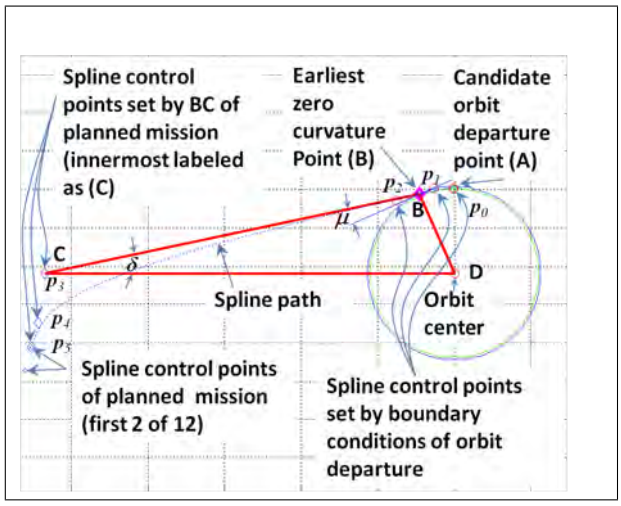

Fig. 5. Geometric construction of feasible path between IC orbit and mission starting point

2, must be adjoined to the planned path to minimize the complexity required to guarantee feasibility. Since loiter patterns can be freely placed, this case is considered to be an exception and will not be further developed here. A direct approach to the planned mission is used to illustrate the general case. Having established a minimum separation bound between or orbit center and the mission start point for each vehicle, it remains to compute the point of departure from the loiter pattern. To determine the departure geometry, we apply the Scheuer graphic construction to determine the point on the augmented turning radius circle for which the flight path direction is pointed towards the solution. We direct the approach from the construction to the innermost control point of the arrival boundary condition, as described by (6), see Figure 5[6]. This geometry[7] permits a relatively straight path following orbit departure but introduces limits on further tailoring for duration. Consequently, we solely use it to identify the *latest* practical orbit departure path after which it may be practical to traverse the orbit again to avoid unnecessary inflection when a path is lengthened to match duration. By selecting a departure earlier than illustrated, the net effect is to permit path lengthening without introducing inflections that tend to result in local maxima of curvature which adversely affect feasibility. We select a departure point whose position angle is rotated further back on the loiter circle from **A** by 75^o to solve (5)[8]. The rationale for this step will be clear when shorter paths are lengthened for simultaneous arrival. The steps of the overall process are listed below (using nomenclature introduced on Figure 5).

[6]The selection of this control point (**C** on Figure 5) to which the path is directed from point **B** is based on the characteristic that the spline path is within the convex hull of control points and in the extreme, the path is actually defined by the control points: for orbit departure from point **A**, the vehicle can achieve straight flight at point **B** with a direction aligned towards point **C**.

[7]The derivation of the geometry shown in Figure 5 is presented in the Appendix.

[8]The path from **A** to **B** is affine so it can be translated/rotated to any position on the orbit.

1) Plan paths off-line for each vehicle using our graph-search based approach [14]. Paths have initial location, direction and curvature as determined by a graph-search based optimization algorithm.
   a) Determine control points $\boldsymbol{p}_3$, $\boldsymbol{p}_4$, and $\boldsymbol{p}_5$ from the boundary conditions at the start of the planned mission.
   b) Based on planned mission, locate points **C** and **D** so that points **A** and **B** may be determined based on orbit around **D** (i.e. locate **D** sufficiently far from $\boldsymbol{p}_5$ as discussed in sub-section II-A).
   c) Determine control points $\boldsymbol{p}_0$, $\boldsymbol{p}_1$, and $\boldsymbol{p}_2$ for a loiter departure point, 75^o earlier on the orbit than $\mathbf{A}'$[9].
2) Augment boundary condition spline solutions so that all paths meet curvature constraints (use tangent scaling).
3) Deploy the vehicles to their staging orbit locations.
4) Determine the duration (length at constant velocity) of a feasible path for each vehicle based on current location, traverse to the departure from orbit point and then to the mission starting point $\boldsymbol{p}_5$.
5) Compare the durations (lengths) of all paths and select the longest as the one to match.
6) Lengthen the shorter paths while abiding by feasibility constraints until all paths have equal duration.
7) Transmit simultaneous arrival path definitions to vehicles and commence mission.

The first three steps above are not time-critical and can be executed as soon as the planner converges. The last four steps are time-critical and constitute the requirements for our simultaneous arrival algorithm. With respect to step 6, since the loiter patterns are asynchronous, the differences in duration can be on the order of the circumference of a loiter orbit. The results of the first two steps are illustrated on Figure 6. The curvature and its derivative with respect to τ for these curves are illustrated on Figure 7. Note, the curvature of these paths far exceeds the feasibility constraints, so they must undergo refinement before time of arrival can be considered.

IV. Augmenting Candidate Paths So That All Are Feasible

In general, a solution, as outlined in Section III-B, determined solely on boundary condition is not feasible as it fails to satisfy curvature constraints, especially near the endpoints, as illustrated above in Figure 7. Therefore, a minimum complexity solution to tailor each spline path for feasibility is required. Additionally, once paths are made feasible, it is preferable that the results not require further feasibility validation when tailoring for simultaneous arrival is carried out. In order to satisfy these objectives and simplify the problem, we constrain the bearing from the orbit center

[9]In case the loiter pattern is not an orbit, the Scheuer construction may be required at the onset from a point between **A** and $\mathbf{A}'$ to establish an equivalent orbit center.

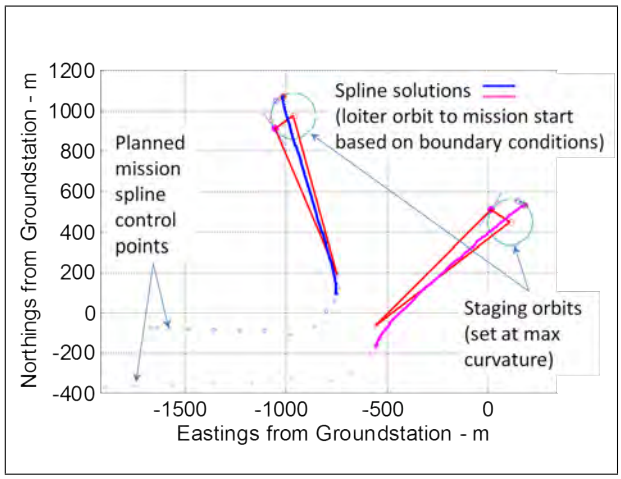

Fig. 6. Candidate paths between IC orbits and mission starting points (showing construction lines to locate orbit departure points)

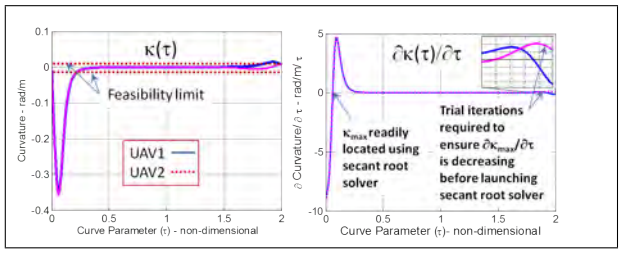

Fig. 7. Curvature and its derivative with respect to the parameter τ for each vehicle (for paths shown on Figure 6)

to the mission start of each path to lie within a $\pm 30^o$ angle of the path's initial tangent vector. This enables use of tangent magnitude scaling to adjust curvature in the vicinity of both endpoints. The six control point spline boundary condition solution has a single degree of freedom at each end with which curvature may be adjusted. G^2 continuity can be preserved by appropriately scaling the first and second derivatives of the path with respect to the parameter, τ. Position and direction of departure (and arrival) are fixed by boundary conditions but the magnitude of the parametric speed and acceleration vectors can be scaled so that direction and curvature is invariant. When the paths are flown, vehicle speed can be set to satisfy C^2 continuity in the time domain, so only G^2 continuity is required spatially. Denoting tangent scaling factors as ρ_{start} and ρ_{end} to be applied respectively to the starting and ending portions of the curve, (5) and (6) can be scaled without changing direction or curvature as follows:

$$\begin{bmatrix} \boldsymbol{p}_0 \\ \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \frac{1}{4} & 0 \\ 1 & \frac{3}{4} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} \boldsymbol{r}(0) \\ \rho_{start}\boldsymbol{r'}(0) \\ \rho_{start}^2\boldsymbol{r''}(0) \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} \boldsymbol{p}_3 \\ \boldsymbol{p}_4 \\ \boldsymbol{p}_5 \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & -\frac{3}{4} & 1 \\ 0 & -\frac{1}{4} & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \rho_{end}^2\boldsymbol{r''}(1) \\ \rho_{end}\boldsymbol{r'}(1) \\ \boldsymbol{r}(1) \end{bmatrix} \tag{8}$$

Curvature, defined below for reference in (9), is a computationally expensive expression in a B-spline representation. Consequently, ensuring peak curvature stays within bounds throughout each path is a challenging optimization problem in this framework. In the general case for the 6 control point structure there will be a local maximum near each endpoint and possibly one more near the midpoint. Rather than incur the computational cost of a precise optimization to find the scale factor, ρ, that minimizes κ_{max} for each path, an approximation of κ_{max} as function of the ρ enables use of a simple minimizing routine that can be executed until κ_{max} is bounded by vehicle turning limits. The first step is to find the parameter, τ, for each candidate curve at which κ_{max} occurs, since we do not want to sample the entire curve to find the maxima. Even though κ and its derivative $\partial\kappa/\partial\tau$ are not polynomials, they are easily expressed as functions of the spline parameter, τ. The structure of the 6 control point boundary condition solution results in $\partial\kappa/\partial\tau$ being a smooth function. The first root of this function with respect to variation in τ from each endpoint identifies the parameter at which κ_{max} occurs on each end. These can be found by using a simple quasi-Newton root solver whose details are described in the Appendix. To guarantee convergence, it is sufficient to require the magnitude of $\partial\kappa/\partial\tau$ for the second iteration to be less than the first, since the spline approaches minimum curvature near it parametric midpoint. In general, a few sample iterations may be required to satisfy this initialization criterion (The inset of the right panel of figure 7 illustrates an example of this at the arrival end). The super-linear convergence rate of this method then permits solution of κ_{max} within as few as five iterations, so computational delay is not a significant issue.

$$\kappa \equiv \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}} = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}} \tag{9}$$

To ensure the spline curves are feasible, iterative values of $\rho_{start(end)}$ are used in conjunction with another instance of the quasi-Newton solver to minimize the magnitude of κ_{max} with respect to $\rho_{start(end)}$ by locating the zeros of $\partial\kappa_{max}/\partial\rho_{start(end)}$. This is accomplished with minimum computation by seeding iterations with small perturbations in ρ with respect to each endpoint[10]. Further application details are provided in the Appendix. For the example shown in Figure 6, the shape of the surface of $\partial\kappa_{max}/\partial\rho$ on which (14) operates is illustrated on Figure 8. The smoothness of the surface illustrates the rationale for selection of such rudimentary solvers. Once all paths have been rendered feasible, shorter ones can be tailored for simultaneous arrival.

V. Determination of Equal Duration Paths (Provide Simultaneous Arrival)

The geometric construction technique described in preceding sub-section permits path duration for each vehicle, from present time to mission start, to be computed as the summation of the time required to traverse along an orbit from present position to the orbit departure point plus the time to exit the orbit and reach the mission start point. The

[10]The 6 control point B-spline does not exhibit local control at each endpoint with respect to ρ but the effect of ρ_{start} on the end portion is second order and vice-versa.

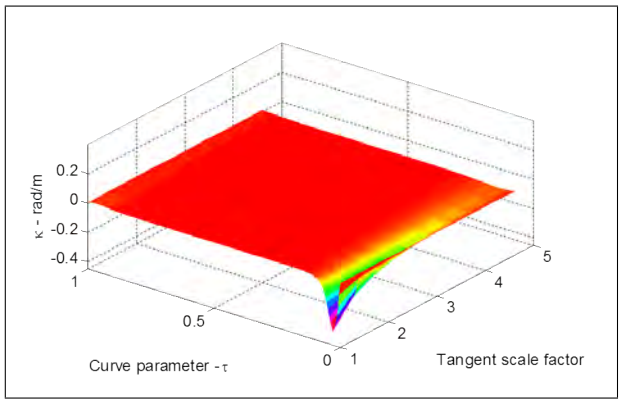

Fig. 8. Curvature variation with tangent scaling applied uniformly at both ends of each curve.

nominal speed of each vehicle is used to relate distance to time. The feasible spline between orbit departure ($\boldsymbol{p}_0$) and mission start ($\boldsymbol{p}_5$) for each vehicle constitutes the portion of each path that can be further tailored for simultaneous arrival. However, if the shorter duration splines are lengthened as defined during the feasibility validation step, they may incur additional local maxima in curvature since tangent scaling can cause control points to crossover in the extreme. In order to preserve the boundary conditions established by the six control points and minimize the effect on curvature, we leverage the local control characteristic of B-splines by subdividing each path and inserting two additional interior control points. We use the knot refinement technique described by Lyche and Mørken in [15] to accomplish this with the following knot vectors:

$$\begin{aligned}&\text{For a knot vector, } t_k, \text{ defined as } t_k \in [0,1]\\&\text{6 control point knot vector } \equiv [0,0,0,0,0,0.5,1,1,1,1,1,1]\\&\text{8 control point knot vector } \equiv\\&[0,0,0,0,0,0.25,0.5,0.75,1,1,1,1,1,1]\end{aligned} \tag{10}$$

The knot insertion process, which yields an identical curve with additional control points, is a matrix-based equation (see [15]), which can be pre-computed so during execution it only requires a single matrix multiply operation. We then tailor the location of the $4th$ control point of the shorter paths to satisfy equal duration (simultaneous arrival) requirements[11]. The path of the latest arrival vehicle is used to establish a specified path length to which the others will be tailored. To effect path lengthening, the $4th$ control point of all short duration paths is progressively perturbed, also using a quasi-Newton solver, along its associated normal line with respect to the feasible path prior to subdivision. The details of finding the path normal associated with the control point are presented in the Appendix. The case introduced in Figure 6 is completed on Figure 9 for the extreme case where the instantaneous position of the vehicle on the left would permit it to instantly depart the loiter orbit while the one on the left was positioned so that it had to complete nearly a full orbit before it could depart its orbit. Hence, the left vehicle path required lengthening, as shown, for simultaneous arrival. The location of the tailored control point and its normal vector to its associated feasible spline is also shown. The

[11]Path lengths are computed using a quadrature-based algorithm we derived in [13]. Nominal speed of each vehicle is used to relate distance to time for simultaneous arrival.

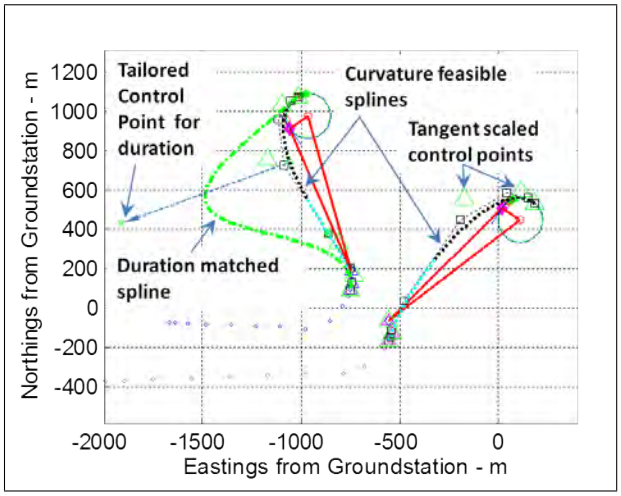

Fig. 9. Completed two vehicle example.

curvature of the tailored paths, both for feasibility through tangent scaling, and path length through adjustment of the $4th$ control point is shown on Figure 10 below. The last

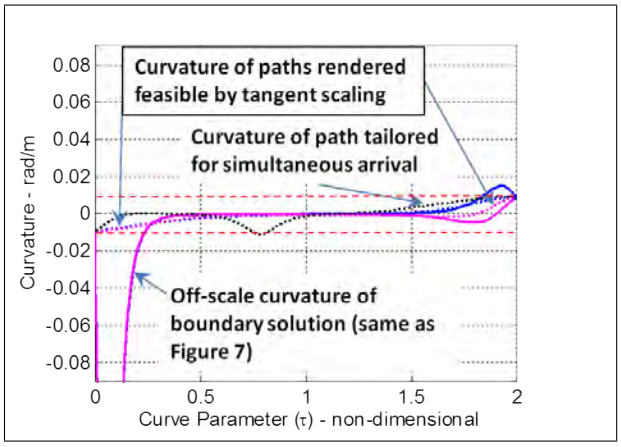

Fig. 10. Curvature with respect to the parameter τ for each vehicle for paths shown on 9

step in this process is to combine the path from current position to orbit departure and the 8 control point splines for each vehicle into a composite which is then approximated by a 12 control point spline, the standard we use for each mission segment for uniform mission specification. The approximation algorithm is identical to that used to define the planned mission segments; see [13] for details.

VI. Flight Test Validation

These concepts were validated during a series of flight trials at the Naval Postgraduate School (NPS) Joint Interagency Field Experiments (JIFX 14-2) held February 10-13,

2014. The NPS UAV lab has developed a Rapid Flight Test Prototyping System (RFTPS) to enable on-board integration of advanced control algorithms from concept to flight test. The principal capability of the system enables advanced control development, integration and deterministic execution of the resulting code on-board, and advanced data logging and communication capabilities. The system is based on tight integration of several software and hardware components including convenient Matlab/Simulink/RTW research and development environment, an advanced industrial autopilot and a high performance microcomputer. Major benefits of the developed RFTPS system include:

- convenience of high level algorithm design, auto-coding capabilities and hard real-time execution,
- simple and transparent interfacing of hardware that is primarily based on the well supported serial and TCP/IP communication,
- inexpensive design of on-board components based on COTS technology,
- advanced payloads, which are highly reliable, using industrial grade PC104 boards and miniature sensors that can be easily changed or upgraded as mission requirements dictate; the telemetry links are secure, yet low power and unobtrusive to the public, thus dispensing with the need for special authorizations from government authorities.

An overview of the system configuration is presented in [13]. A key feature of this system is the integration of all payload subsystems via Ethernet. On-board the vehicle, subsystems are tied to a mesh access point through which data are communicated to the ground. The IP-based architecture makes the system scalable to any sensor configuration that can be carried by the vehicle and provides open and secure access. Moreover, the self-configuring wireless mesh network data communication (wave relay) is independent from the primary autopilot command/control link for safety of flight. The mission sensors used for this research include full motion video (FMV) and still cameras that are pointed by gimbal systems with one (roll stabilized) or two degrees of freedom (pan-tilt for direct pointing). The type of vehicle we use is shown in Figure 11. The NPS mobile ground station from which experiments are conducted is shown in Figure 12.

Fig. 11. NPS Flight Test Vehicle

Fig. 12. NPS Mobile GCS

A. *Flight test results*

Figure 13 depicts flight test data for the approach errors of a single vehicle when a planned path is approached from arbitrary loiter orbit locations and the coordinated arrival algorithm is *not* used. In these cases the autopilot converges to precise trajectories once the line orthogonal to the initial point is breached but entry is not precise, nor is the path taken of consistent duration. This can degrade coordination across the team and cause a portion of the planned path to be missed. The coordinated arrival algorithm is designed to permit all vehicles to achieve simultaneous arrival with C^2 continuity at the initial points of their planned paths. Flight test results for our algorithm for a variety of starting

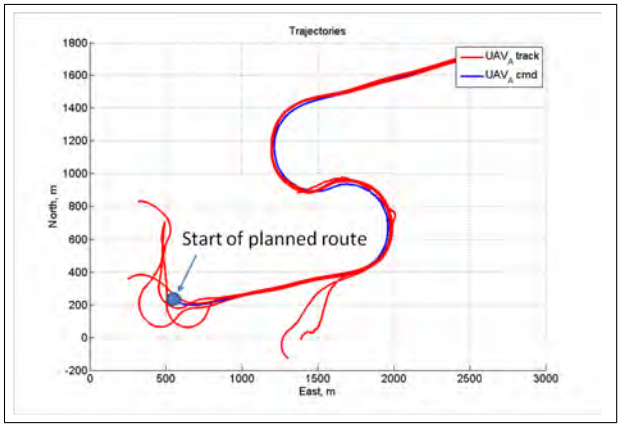

Fig. 13. Vehicle tracks to start of a planned mission from arbitrary orbit locations (coordinated arrival *not* used.

locations are presented on Figure 14. All flight path tracks join their respective planned paths with C^2 continuity at the entry points with precise coordination in time. These paths enable coordinated missions to commence as planned.

VII. Conclusions

We have demonstrated a geometrically derived algorithm to compute simultaneous arrival paths for a team of fixed-wing aerial vehicles with fixed boundary conditions and curvature constraints along the path. By dividing the problem into the stages of finding a boundary condition solution

Fig. 14. Coordinated vehicle tracks to start of a planned mission from arbitrary orbit locations (acquired from flight test).

for each vehicle, tailoring these for feasibility, and then lengthening all but the longest in duration, the sub-problems admitted elementary geometric solutions. The bounds on these solutions were empirically derived to enable a complete orbit circumference to be added to the shortest path. We selected quasi-Newton root solvers for their super-linear rate of convergence. The application of parametric polynomials rendered the intricate functionals that must to be optimized within constraints as very smooth curves and surfaces. This permitted the partial derivatives required for optimization to be computed using simple difference equations. By minimizing the number of control points used for path representation, the presence of local minima were addressed without issue. The result is computationally simple enough to for near-real time execution. Precise team coordination can be important for surveillance type missions. This algorithm is also applicable to other types of vehicles that can be characterized as drift-less. As derived, this approach requires relatively calm winds to be successful. Steps to provide robustness to winds are ongoing but results from flight test indicate the current approach is practical.

VIII. Future Work

We are in the process of generalizing the algorithm to support mission approach from any direction and at any proximity. As described, if the initial tangent vector of the planned path is opposite to the direction from which it is approached, this algorithm can fail to sufficiently tailor curvature at the arrival end because it relies on tangent scaling. As noted earlier, a further instance of appending a Scheuer construction on the arrival end is a practical approach to resolve this issue but the bounds for its application are still being derived. Furthermore, we are looking at ways to make the algorithm more robust to winds. In this regard, we are considering ways to iteratively call the algorithm if wind or other disturbances introduce errors in the coordinated spline paths. A simpler version of the algorithm can be used once a vehicle is directed towards its initial planned mission point.

APPENDIX

A. Determining a practical loiter orbit departure point

Parameters for the geometry presented in Figure 5 can be determined as follows: the orbit center, point **D**, is set by the user based on an offset from the mission start as described earlier and illustrated earlier in Figure 3. Therefore, distance $\bar{\mathbf{CD}}$ is known. The angle $\angle\mathbf{CBD}$ is also known: $\angle\mathbf{CBD} = \mu + \frac{\pi}{2}$, based on Scheuer's construction. The Law of Sines can then be used to find the angle denoted as $\angle\delta$, since its corresponding line segment is the radius of the augmented turning circle. This relation is used to locate point **B** and hence **A** for each vehicle's orbit.

B. Finding local maxima in curvature

To find the parameter, τ, such that $\kappa(\tau)$ is a local maximum of a 6 control point spline, we look for roots of $\partial\kappa/\partial\tau$. A quasi-Newton method[12] can be set up to find these roots by sampling perturbations in τ from each end such that $\kappa'_2 < \kappa'_1$ for τ_2 more interior to the curve than τ_1. Successive iterations may be found:

$$\tau_n = \tau_{n-1} - \kappa'(\tau_{n-1}) * \frac{\tau_{n-1} - \tau_{n-2}}{\kappa'(\tau_{n-1}) - \kappa'(\tau_{n-2})} \tag{11}$$

The structure of the 6 control point spline permits this technique to be started from each end to rapidly provide the local maxima of a given path so that it can be tailored if required to satisfy feasibility constraints.

C. Tangent scaling to tailor curvature of interior portions of a path

The use of tangent vector scaling at each end of a path to ensure $\kappa_{max}(\rho)$ satisfies feasibility constraints is a multivariate optimization problem, i.e. at each end: $\kappa_{max_{\text{start(end)}}} = f(\rho_{start}, \rho_{end})$, where the values at each end are calculated from small perturbations about a central value[13]. Approximating $\kappa_{max}(\rho_{start}, \rho_{end})$ at each end by a second order Taylor Series yields:

$$\kappa_{max}(\rho) \approx \kappa_{max}(\rho_i) + \langle \nabla\kappa_{max}(\rho_i), (\rho - \rho_i)\rangle + \frac{1}{2}(\rho - \rho_i)D^2\kappa_{max}(\rho_i)(\rho - \rho_i) \tag{12}$$

for $\rho \in \Re^2$: $(\rho_{start}, \rho_{end})$

Where ρ_i denotes the central values for the ith iteration about which the series is evaluated. If the Hessian matrix, $D^2\kappa_{max}(\rho_i)$, is positive definite, iterative values to find a minimal $\kappa_{max}(\rho)$ can be found:

$$\rho_{n+1} = \rho_n - [D^2\kappa_{max}(\rho_n)]^{-1}\nabla\kappa_{max}(\rho_n) \tag{13}$$

The quasi-Newton approach is to approximate the Hessian through difference equations. Furthermore, by noting that its off-diagonal terms are of second order significance, the equation can be diagonalized so a matrix inverse calculation is not required. Iterations are alternated for ρ_{start} and ρ_{end} with updates only made to ρ_{start} when the nth iteration of $\kappa_{max}(\rho_{start})$ fails bounds on the starting portion and to ρ_{end}

[12]The classic Newton-Raphson iterative root solver can be expressed: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$. Quasi-Newton methods approximate the derivative, typically through difference equations. While they do not converge as rapidly as Newton-Raphson, convergence is still super-linear and may require less computational time, especially when derivatives are costly to compute as in the case of curvature.

[13]See Equation (14) for nomenclature.

when the nth iteration of $\kappa_{max}(\rho_{end})$ fails bounds on the ending portion. The process is initiated by using a pair of values of at $\rho_{start(end)}$ that constitute a small perturbation to find numerical expressions for $\partial\kappa_{max}/\partial\rho$ at each end of the spline. These are used to seed this application of the solver to optimize ρ until the magnitude of κ_{max} is within feasibility bounds. Again, super-linear convergence means only a few iterations are required to tailor the six control point spline curve for each vehicle until it is rendered feasible. In general, the final values of the scale factors are not equal ($\rho_{start} \neq \rho_{end}$), so the separate scaling is required if such a simple technique is to be successful. The equations for this process are described below in (14).

$$\begin{gathered}\frac{\partial\kappa_{max}}{\partial\rho_i} \approx \frac{\kappa_{max}\left(\rho_b\right) - \kappa_{max}\left(\rho_a\right)}{\rho_b - \rho_a} \\ \rho_i = \frac{\rho_b + \rho_b}{2}\end{gathered} \tag{14}$$

Where $\rho_a = 1$ and $\rho_b = 1 + \epsilon$ for $i = 1$; $\rho_a = 1 + \delta$ and $\rho_b = 1 + \delta + \epsilon$ for $i = 2$. For iterations where $i > 2$, ρ_i is then set by the quasi-Newton solver:

$$\rho_i = \rho_{i-1} - \kappa_{max_{i-1}} \frac{\rho_{i-1} - \rho_{i-2}}{\kappa_{max_{i-1}} - \kappa_{max_{i-2}}} \tag{15}$$

Perturbations are then set by: $\rho_a = \rho_i - \epsilon/2$ and $\rho_b = \rho_i + \epsilon/2$; for $\delta << 1$ and $\epsilon << \delta$.

D. Finding the normal vector from a curve to a (control) point

The normal vector from a spline curve to one of its control points can be found by noting that the inner product of the tangent at that point on the curve with respect to the vector from the curve, $\mathbf{r}(\tau) = [x(\tau), y(\tau)]$, to the (control) point, $\mathbf{p}_i = [x_i, y_i]$, is zero there:

$$\begin{gathered}\left\langle \frac{d\mathbf{r}}{d\tau}(\tau), (\mathbf{p}_i - \mathbf{r}(\tau) \right\rangle = 0.0 = \\ \frac{dx}{d\tau}(\tau) \cdot (x_i - x(\tau)) + \frac{dy}{d\tau}(\tau) \cdot (y_i - y(\tau))\end{gathered} \tag{16}$$

A rapidly converging iterative relation can be set up to find the parameter, τ, for which a control point, $\mathbf{p}_i$, is normal to the curve by leveraging the shape of the basis functions. By starting a Newton-Raphson iteration at the value of τ associated with the maximum of the ith basis function, the user can set the desired tolerance on the magnitude of the inner product. This relation can be used to identify the normal vector of a curve that intersects a designated control point. We use this approach to identify the line along which the fourth control point in the 8 point spline can be slid to lengthen a path. By using this vector, the curvature is not increased until a path has been significantly lengthened and the curve is locally governed by a single basis function.

ACKNOWLEDGMENT

We would like to thank the Naval Postgraduate School's field experimentation team for hosting their Joint Interagency Field Exploration (JIFX) events at Camp Roberts, CA, through which these algorithms have been validated. We would also like to thank the U.S. Navy for permitting these tests to be executed.

REFERENCES

[1] A. Richards, J. Bellingham, M. Tillerson, and J. P. How, "Coordination and control of multiple UAVs," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, Monterey, CA, August 2002, AIAA Paper 2002-4588.

[2] E. Xargay, V. Dobrokhodov, I. Kaminer, A. M. Pascoal, N. Hovakimyan, and C. Cao. (2011) Time-critical cooperative control for multiple autonomous vehicles. Calhoun Institutional Archive of the Naval Postgraduate School. [Online]. Available: http://hdl.handle.net/10945/36038

[3] V. Cichelle, I. Kaminer, E. Xargay, V. Dobrokhodov, N. Hovakimyan, P. A. Agular, and A. M. Pascoal, "A lyapunov-based approach for time-coordinated 3d pathfollowing of multiple quadrotors," in *Proceedings of the 51st IEEE Conference on Decision and Control*, Maui, Hawaii, Dec. 10, 2012, pp. 1776–1781.

[4] H. Delingette, M. Hébert, and K. Ikeuchi, "On the generation of nearly optimal, planar paths of bounded curvature and bounded curvature gradient," in *Proceedings of the International Conference on Intelligent Robots and Systems*, Osaka, Japan, November 1991.

[5] D. R. Nelson, T. W. McLain, and R. W. Beard, "Experiments in cooperative timing for miniature air vehicles," *Journal of Aerospace Computing, Information, and Communication*, vol. 4, no. 8, pp. 956–967, 2007.

[6] H.-S. Kim and O. Cheong, "The cost of bounded curvature," *Computing Geometry*, vol. 46, no. 6, pp. 648–672, 2013.

[7] A. Scheuer and T. Fraichard, "Continuous-curvature path planning for car-like vehicles," in *Proceedings of IEEE-RSJ International Conference on Intelligent Robots and Systems*, Grenoble, FR, September 1997, pp. 997–1003, volume 2.

[8] A. Scheuer and C. Laugier, "Planning sub-optimal and continuous-curvature paths for car-like robots," in *Proceedings of IEEE-RSJ International Conference on Intelligent Robots and Systems*, Victoria, BC, CA, October 1998, pp. 216–222.

[9] E. Bakolas and P. Tsiotras, "On the generation of nearly optimal, planar paths of bounded curvature and bounded curvature gradient," in *Proceedings of the American Control Conference*, St. Louis, MO, October 2009, pp. 385–390.

[10] M. Shanmugavel, "Path planning of multiple autonomous vehicles," Ph.D. dissertation, Cranfield University, Cranfield, Bedford, UK, 2007.

[11] E. Seckel, *Stability and Control of Airplanes and Helicopters*. New York, NY: Academic Press, 1964.

[12] R. Stengel, *Flight Dynamics*. Princeton, NJ: Princeton University Press, 2004.

[13] J. Keller, D. Thakur, V. Dobrokhodov, K. Jones, M. Pivtoraiko, J. Gallier, I. Kaminer, and V. Kumar, "A computationally efficient approach to trajectory management for coordinated aerial surveillance," *Unmanned Systems*, vol. 1, pp. 59–67, June 2013.

[14] D. Thakur, M. Likhachev, J. Keller, V. Kumar, V. Dobrokhodov, K. Jones, J. Wurz, and I. Kaminer, "Planning for opportunistic surveillance with multiple robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, November 2013, pp. 5750–5757.

[15] T. Lyche and K. M. rken, *Spline Methods, Draft*. Oslo, Norway: Department of Informatics, Centre of Mathematics for Applications, University of Oslo, 2011, p. 88.