

Toward n-Ship Computation of Trajectories for Shared Airspace

Daniel D. Moerder*

Paul M. Rothhaar†

This paper considers an approach for modelling transport aircraft trajectories that can facilitate their rapid evaluation and modification, either enroute or in terminal control areas, with the goal of efficiently making use of airspace and runways by a large population of vehicles without pairwise violation of separation criteria.

I. Introduction

Although it is hoped that, at some point in the future, the National Airspace System (NAS) will be crowded everywhere, at present it is most crowded in the vicinity of airports; that is, in the Terminal Radar Approach Control – TRACON airspace. Critical elements of terminal area traffic management are scheduling aircraft to runways and satisfying constraints on separation between aircraft in the TRACON.

The accuracy and reliability of the scheduling process has an important impact on aircraft operating cost and environmental impact. In an ideal world, where each aircraft’s Estimated Time of Arrival (ETA) would be exact, approaching aircraft could make adjustments to their trajectories to achieve Scheduled Times of Arrival (STAs) assigned by TRACON while at cruise altitude, then descend to land precisely on time. This would minimize time spent in noisier, less fuel-efficient flight at low altitude, and would minimize time between landings. Instead, aside from the fact that disruptive events, such as runway closures, occur during operations, ETAs meet STAs with less than perfect accuracy. Because of this, as noted in Section 5 of¹

While traffic-management procedures in use today generally achieve smooth traffic flow even when delays have built up significantly, controversy lingers over what is the best procedure for delay absorption. The dividing chasm is between pilots and airline operators on the one hand and controllers and traffic managers on the other.

Pilots and airline operators prefer delays to be absorbed close to the airport, even to the point where holding is required in the TRACON airspace at low altitude. They fear that early delay absorption far from the airport does not produce sufficient traffic pressure to achieve a high landing rate.

Traffic managers and controllers, on the other hand, contend that, on balance, it is more efficient to absorb most delay in the Center airspace far from the airport so as to maintain traffic flow in the TRACON smoothly and orderly. They further contend that delay-absorption strategies that lead to frequent holding in the TRACON airspace create high workload for controllers and risk chaotic traffic conditions that actually reduce landing rates.

The advanced state of the art in terminal area control is the Center/TRACON Automation System (CTAS), jointly developed by NASA Ames Research Center (ARC) and the Federal Aviation Administration (FAA)^{12,3} CTAS comprises a set of algorithms that provide advisories and options to assist human air traffic controllers in managing traffic to safely make efficient use of runways.

Both of these functions use CTAS-executed simulations of the vehicles. In the case of runway scheduling, the simulation provides estimated time of arrival (ETA); in the case of separation, simulated trajectories are compared to detect conflict.

*Senior Research Engineer, NASA Langley Research Center, Hampton, VA

†Research Engineer, NASA Langley Research Center, Hampton, VA

The simulation, described in,² is strongly optimized for speed of execution, using a highly simplified point mass dynamical model. The simulation draws on radar and transponder data for periodically updated trajectory initial conditions, and hosts vehicle-specific engine and drag models for a number of different aircraft, basing its selection on the vehicle type declared in the aircraft’s flight plan. Trajectory guidance is procedural. This guidance approach is chosen to speed execution when compared with a full-up trajectory optimization, and it also produces trajectory commands which can be directly and clearly relayed to a pilot. When two TRACON trajectories exhibit Loss of Separation (LoS) – violation of pairwise horizontal and vertical minimum distances mandated by the Federal Aviation Administration (FAA)⁴ – one or more alternate trajectories are computed and compared with the set of other TRACON trajectories until all trajectories produce satisfactory arrival time estimates without LoS.

Is it possible to improve on this? Certainly, but it would require substantially more communication between the aircraft and TRACON controller, and would strongly benefit from having each aircraft compute and follow its own 4D path to the assigned runway and STA. Recognizing that CTAS has already demonstrated an ability to strongly improve arrival accuracy⁵ when compared to fully manual air traffic control, its main conceptual disadvantage is the lack of a tightly coupled feedback of ETA error to updates in the approach trajectory, which could be mitigated by placing the approach guidance process in the aircraft, rather than in the control tower.

Clearly, a process in which two or more “lone cowboys” individually compose and follow trajectories onto a runway without supervision invites LoS. The role of the traffic controller in this paradigm, should not only include assignment of STAs and runways, but should include modifying the individual approach plans to avoid LoS. We envision the following process:

1. Aircraft communicates its earliest arrival time to TRACON, while in cruise.
2. TRACON assigns a runway and STA, possibly other trajectory constraints, as appropriate, *e.g.* entrance conditions for flying in a landing or holding pattern.
3. Aircraft computes a 4D trajectory to assigned conditions.
4. TRACON evaluates the trajectory for LoS and “bends it and others a little” if necessary, communicating the modified non-LoS trajectory or trajectories back to the affected aircraft.
5. Aircraft recomputes its 4D trajectory to comply with the shape of the TRACON-commanded trajectory, and flies in.
6. All parties repeat steps 2-5 as necessary.

There are several items of technology that are necessary for this process to work. These are:

- a means of communicating proposed and modified trajectories between aircraft and ground,
- a fast means of checking all of the trajectories in TRACON airspace for LoS, and for “bending” them as necessary,
- onboard guidance for composing and following a 4D trajectory, and modifying it in response to TRACON input.

The first item would be facilitated by a trajectory representation that captures relevant detail for transport aircraft trajectories while using a fairly small number of parameters. This could serve to reduce communication overhead.

The second item could, potentially, be realized by a heuristic pairwise LoS detection and trajectory modification approach, but an optimization-based approach in which multiple trajectories are simultaneously modified to eliminate LoS while minimizing deviation from the aircraft-proposed trajectories would be preferable, since limiting changes in the commanded trajectories is beneficial for pilot and controller workload. In order to use efficient gradient-based optimization techniques, it is necessary that the representation, and calculations using it, should be differentiable. This includes calculation of discrete events, such as points of closest approach between trajectories.

In this paper, we examine quadratic splines as a candidate trajectory representation. In Section 2, their structure and advantages are described, and necessary conditions are derived for fitting such splines to

tabulated trajectories with minimum sum-of-squares error. A very simple algorithm is described for selecting the spline’s temporal structure. Section 3 explores a numerical example that demonstrates the approach. Section 4 presents conclusions and speculates about future work.

II. Quadratic Splines

Quadratic splines $q(t)$ are piecewise-quadratic functions (PQF) that satisfy continuity conditions on q and \dot{q} at “knots” – instants t_k that join a $k - 1^{th}$ quadratic arc to a k^{th} one, to form a smooth curve. As PQFs, these splines have attractive features for representing and terminal area trajectories.

- Acceleration in quadratic splines is piecewise constant. Transport aircraft trajectories are, ideally, boring, joining unaccelerated cruise legs with interludes of fairly constant accelerations. The functional form of quadratic splines, then, is a good match for what actually happens in the trajectories they model.
- The square of distance between two trajectories – that is, the distance between locations at a given instant – represented by quadratic splines is piecewise-quartic in time. This enables finding points of closest approach via closed-form, non-iterative expressions. The quartic’s first derivative is cubic, and its second derivative is quadratic. Minima internal to a given piecewise-quartic arc occur where the first derivative is zero, and the second derivative is positive. Since the cubic has real coefficients, there will be either one or three roots, one of which will be relevant. This means that points of closest approach can be determined without time-consuming exploration of pairs of simulated trajectories; instead, simple algebraic expressions are evaluated on each subarc. This particularly simplifies LoS constraints when using quadratic splines in an optimization.
- Anecdotal experience indicates that functional simplicity in a model is beneficial when posing numerical optimization problems. A quadratic spline is much less complicated and nonlinear than a point-mass transport aircraft dynamical model. It might be objected that, if used in the optimization problem described above to modify a given trajectory, there’s no guarantee that the resulting modified trajectory would be comfortably flyable, since the aircraft dynamics are not included in the spline. In response to that, first, approach trajectories are flown well within the interior of the aircraft’s flight envelope, so that neighboring trajectories should also be flyable; second, the optimization performance goal is to minimize the amount of modification experienced by each trajectory. This also contributes to confidence that the optimized trajectory would be flyable.

The time history of a given terminal area trajectory over a time interval $a \leq t \leq b$ is represented by a quadratic spline $q(t)$, given in (1), for each of three orthogonal coordinates, say for example, North-East-Down (NED):

$$q(t) = \begin{cases} q_0 = \alpha_0 + \beta_0 \delta t_0 + \gamma_0 (\delta t_0)^2 & a \leq t \leq \bar{t}_1 \\ q_1 = \alpha_1 + \beta_1 \delta t_1 + \gamma_1 (\delta t_1)^2 & \bar{t}_1 \leq t \leq \bar{t}_2 \\ \vdots & \vdots \\ q_N = \alpha_N + \beta_N \delta t_N + \gamma_N (\delta t_N)^2 & \bar{t}_N \leq t \leq b \end{cases} \quad (1)$$

where

$$\delta t_k = t - \bar{t}_k, \quad k = 0, \dots, N \quad (2)$$

The conditions defining the spline in (1) are as follows. Denote, for a given coordinate, the position and velocity at times a and b as $\sigma_a, \dot{\sigma}_a$ and $\sigma_b, \dot{\sigma}_b$, so that

$$\begin{aligned} q_0(0) &= \sigma_a & q_N(b - \tau_N) &= \sigma_b \\ \dot{q}_0(0) &= \dot{\sigma}_a & \dot{q}_N(b - \tau_N) &= \dot{\sigma}_b \end{aligned} \quad (3)$$

Further, the spline satisfies position and velocity continuity conditions at the knots; that is, the instants that join quadratic arcs of q :

$$q_k(\tau_k) = q_{k+1}(0), \quad \dot{q}_k(\tau_k) = \dot{q}_{k+1}(0), \quad k = 1, \dots, N \quad (4)$$

where

$$\tau_1 = \bar{t}_1 - a, \quad \tau_k = \bar{t}_k - \bar{t}_{k-1}, \quad k = 2, \dots, N \quad (5)$$

The $\alpha_k, \beta_k, \gamma_k$, coefficients in $q(t)$ are selected by fitting it to a tabulated trajectory $\bar{\sigma}$:

$$\hat{t} = \begin{bmatrix} \hat{t}_1 = a \\ \hat{t}_2 \\ \vdots \\ \hat{t}_m = b \end{bmatrix}, \quad \bar{\sigma} = \begin{bmatrix} \sigma_a \\ \sigma(\hat{t}_2) \\ \vdots \\ \sigma_b \end{bmatrix} \quad (6)$$

where $m \gg N + 1$. For a given distribution of τ_k from (5), seek to minimize

$$\mathcal{J} = \sum_{k=0}^N \sum_{j=\ell_k^-}^{\ell_k^+} e_j^2, \quad e_j = q_k(\hat{t}_j - \bar{t}_k) - \sigma_j \quad (7)$$

$$\hat{t}_{\ell_k^- - 1} \leq \bar{t}_k \leq \hat{t}_{\ell_k^-}, \quad \hat{t}_{\ell_k^+} \leq \bar{t}_{k+1} \leq \hat{t}_{\ell_k^+ + 1}, \quad \ell_{k-1}^+ = \ell_k^- - 1 \quad (8)$$

Equation (7) can be simplified by grouping the coefficients for q_k as

$$\omega_k = \begin{bmatrix} \alpha_k & \beta_k & \gamma_k \end{bmatrix}^T \quad (9)$$

so that

$$\begin{aligned} q_k(t) &= T(t - t_k)\omega_k, & T(\delta) &= \begin{bmatrix} 1 & \delta & \delta^2 \end{bmatrix} \\ \dot{q}_k(t) &= D(t - t_k)\omega_k, & D(\delta) &= \begin{bmatrix} 0 & 1 & 2\delta \end{bmatrix} \end{aligned} \quad (10)$$

The cost, then, is

$$\mathcal{J} = \sum_{k=0}^N \bar{e}_k^T \bar{e}_k, \quad \bar{e}_k = M_k \omega_k - \bar{\sigma}_k \quad (11)$$

$$M_k = \begin{bmatrix} T(\hat{t}_{\ell_k^-} - \bar{t}_k) \\ T(\hat{t}_{\ell_k^- + 1} - \bar{t}_k) \\ \vdots \\ T(\hat{t}_{\ell_k^+}) \end{bmatrix}, \quad \bar{\sigma}_k = \begin{bmatrix} \sigma_{\ell_k^-} \\ \sigma_{\ell_k^- + 1} \\ \vdots \\ \sigma_{\ell_k^+} \end{bmatrix} \quad (12)$$

and the constraints (3, 4) are rewritten

$$B_k = \begin{cases} \sigma_a - T(0)\omega_0 & k = 0 \\ T(\tau_{k-1})\omega_{k-1} - T(0)\omega_k & k = 1, \dots, N \\ \sigma_b - T(b - \bar{t}_N)\omega_N & k = N + 1 \end{cases} \quad (13)$$

$$C_k = \begin{cases} \dot{\sigma}_a - D(0)\omega_0 & k = 0 \\ D(\tau_{k-1})\omega_{k-1} - D(0)\omega_k & k = 1, \dots, N \\ \dot{\sigma}_b - D(b - \bar{t}_N)\omega_N & k = N + 1 \end{cases} \quad (14)$$

Note that, in our study, the σ rates at the initial and terminal points are estimated by first-order differences. With (11,13,14) in hand, the optimization problem's Lagrangian is written as

$$\mathcal{L} = \mathcal{J} - \sum_{k=0}^{N+1} (\lambda_k B_k + \eta_k C_k) \quad (15)$$

and necessary conditions for optimality are obtained by differentiating \mathcal{L} with respect to the ω_k, λ_k , and η_k , and setting the derivatives to zero. This results in a linear system of dimension $3(N + 1) + 2(N + 2)$.

With the least-squares-optimal quadratic spline in hand for a given pattern of knots, *i.e.*, $\bar{t}_k, k = 1, \dots, N$, the next question is, what is the best pattern of knots for the dataset $\{\hat{t}, \bar{\sigma}\}$? The answer is partially determined by the fidelity to the dataset required of the spline, and partially by the degree to which the motion in the dataset differs from the piecewise constant accelerations of the spline. Two acceleration-related phenomena merit placement of a knot:

1. when acceleration varies sufficiently from an existing knot “ k ” to a point in the data, *i.e.*

$$|\ddot{\sigma}_k - \ddot{\sigma}_j| \geq \text{A Threshold}, \quad j > k$$

2. when the change in slope due to acceleration is sufficiently inconsistent between an existing knot and another point, *i.e.*

$$|(\ddot{\sigma}_k - \ddot{\sigma}_j)|(t_j - t_k) \geq \text{Another Threshold}, \quad j > k$$

The algorithm described below was used in computing the knot distributions that appear in Section 3, and is based on a numerical estimate of the data’s first and second derivatives, which will be referred to as $\bar{\sigma}'$ and $\bar{\sigma}''$, respectively, in the sequel. The two thresholds mentioned above are, for acceleration variation,

$$T_a = \epsilon_a \left(\max_{j=1, \dots, m} \sigma_j'' - \min_{j=1, \dots, m} \sigma_j'' \right) \quad (16)$$

and, for slope variation,

$$T_s = \frac{\epsilon_s}{m} \sqrt{\sum_{j=1}^m (\bar{\sigma}')_j^2} \quad (17)$$

where ϵ_a and ϵ_s are user-selected positive constants that serve as “tuning knobs” for controlling the placement of knots. The knot placement algorithm is applied to m values of $\bar{\sigma}$ data, indexed as $j = 1, \dots, m$. Knot locations are indexed with k , starting with $k = 1$.

1. $k = 1, \quad \tau = []$, $dd\sigma_0 = \bar{\sigma}_1''$, $k_0 = 1$, $k_1 = 1$
2. $k_1 = k_1 + 1$, $dt = \hat{t}_{k_1} - \hat{t}_{k_0}$, $\delta_{dd} = |\bar{\sigma}_{k_1}'' - dd\sigma_0|$
3. IF $\delta_{dd} > T_a$ OR $dt * \delta_{dd} > T_s$ THEN
 - $k_1 = \max\{k_1, k_0 + 1\}$, $k_0 = k_1$, $dd\sigma_0 = \bar{\sigma}_{k_1}''$
 - $k = [k \quad k_1]$, $\tau = [\tau \quad dt]$
4. IF $k < m$ GO TO 2
5. IF $k_1 < m$ THEN $k = [k \quad m]$

Note that the k and τ arrays are lengthened by concatenation in step 3, and k in step 5. That notation is directly parsed in MATLAB as concatenation of a scalar to a single-row array.

III. Numerical Experiments

This Section evaluates the performance of quadratic splines in approximating an IFR flight dataset recorded from the DFW TRACON. This dataset is an approach and landing from eighteen minutes out, recorded every one or five seconds, and resampled to 1/3 hertz, for a total of 360 datapoints in each axis.

The original dataset was converted from latitude, longitude, and altitude to NED coordinates with origin located at the end of the runway, then very lightly filtered to remove some of the radar-sourced noise. Initial and terminal rates for spline boundary conditions were estimated via first-order differences. Figure 1 displays the filtered, NED data for the horizontal motion and the vertical time history of the approach in its left and right plots, respectively.

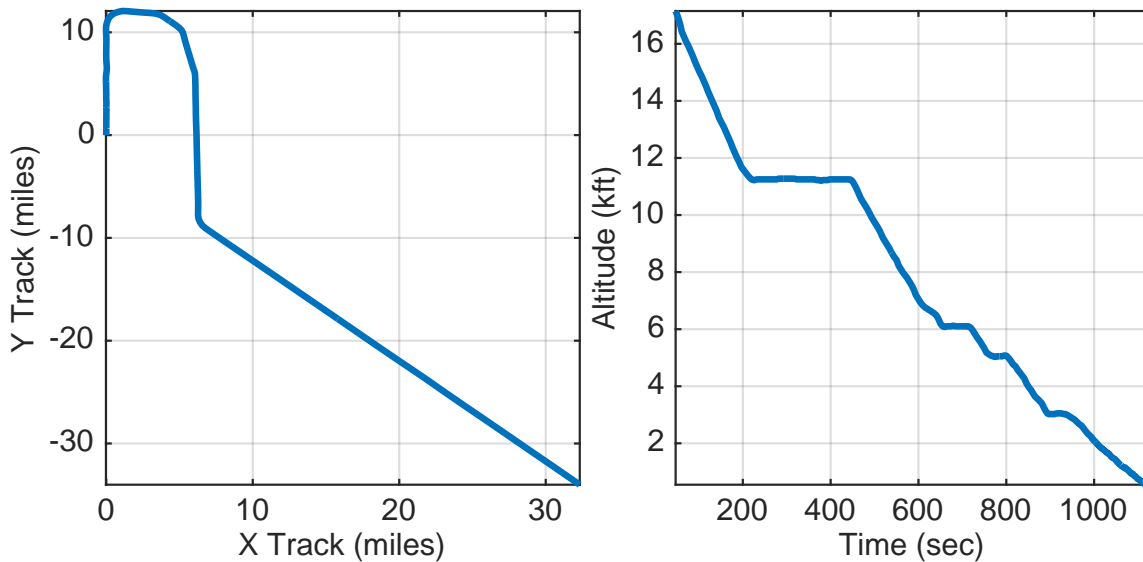


Figure 1. Aircraft Trajectory Data

Before proceeding further, remember that this approximation technique would almost certainly not be used to approximate this type of data. TRACON is interested in an aircraft’s motion from the current instant until it lands or exits TRACON airspace. The quadratic splines are meant to provide a compact means for communicating, comparing, and manipulating intended, rather than historical trajectories. The difference between intended and actual flight is strongly marked in the altitude history on the right of Figure 1. This profile is, frankly, more gnarled than would be deliberate for any measure of desired performance, particularly during 600-900 seconds.

The tabulated time histories in each of the three NED coordinates were fit with splines having several different knot distributions, resulting from different selections of ϵ_a and ϵ_s in the knot locating algorithm described at the end of Section 2. These tuning parameters were varied to obtain two-parameter families of quadratic spline fits. Note that this process was entirely non-iterative. An acceptable approximation reliably results from a fast, though brute-force, process.

Figures 2, 3, and 4 display the variation of maximum error and the numbers of knots as functions of ϵ_a and ϵ_s for splines approximating the x -direction, y -direction, and z -direction time histories. Although the selection process stepped through a full rectangular array of ϵ_a and ϵ_s values, errors and knot numbers are only shown for values that resulted in practically relevant splines. In particular, for the horizontal-plane Figure 2 and 3, data were eliminated where maximum error was greater than 0.5 miles or the number of knots was greater than 50. For the altitude surfaces in Figure 4, the rejection criterion was max error greater than 250 feet, or more than 50 knots. Typically, large values of ϵ_a , ϵ_s resulted in errors that were excessively large, and very small values of ϵ_a , ϵ_s resulted in accurate fits that required too many knots.

It is worth noting that the error and knots surfaces of Figure 4, for the z trajectory, are substantially more chaotic looking than those in Figures 2 and 3. This wild variation, with many isolated peaks (that were filtered out of the plots) is a result of the very wrinkly nature of that trajectory data. Again, these splines are not meant for approximating “noisy” data, though it will be seen that they do an effective job of it.

Figures 5, 6, and 7 display the splined horizontal-plane trajectories for the case where maximum error was held below one mile, 0.2 mile, and 0.1 mile, respectively. In each case, the spline selected was the one which matched the error criterion with the fewest knots. In each Figure, the radar track is shown in blue and the spline fit is shown as a red dashed line. The x and y spline knots are shown as green circles and black triangles, respectively. The trends in the plots are not surprising, with more knots needed to achieve tighter accuracy, but note that only a very modest number of knots are required to achieve the fit – between 6 and 18, depending on accuracy. This probably represents a significant advantage in communicating trajectories between aircraft and TRACON. The number of numbers required to communicate the spline is

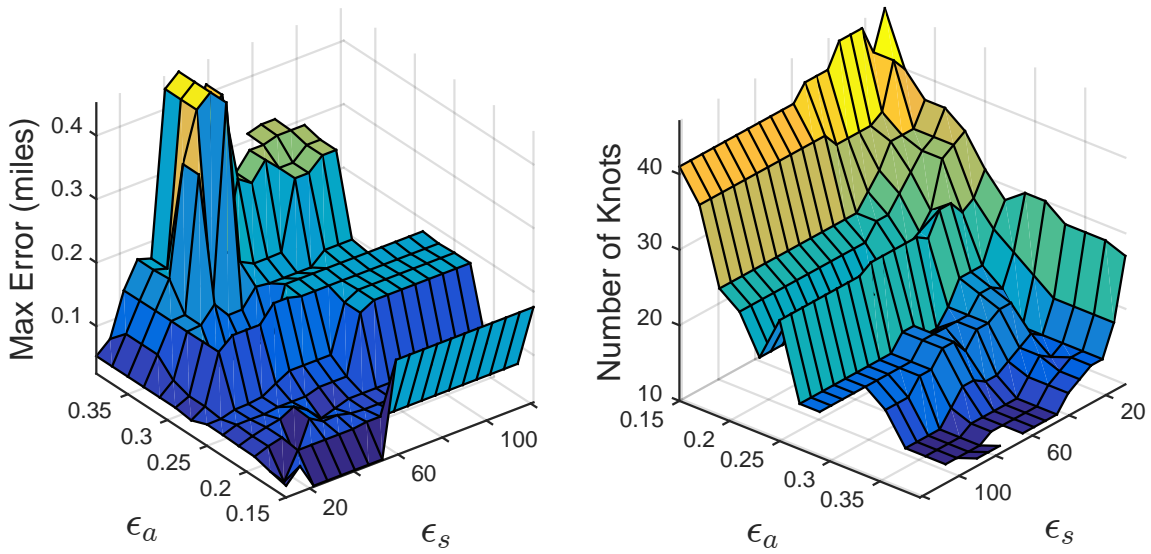


Figure 2. Spline Max Error and Number of Knots for x Trajectory

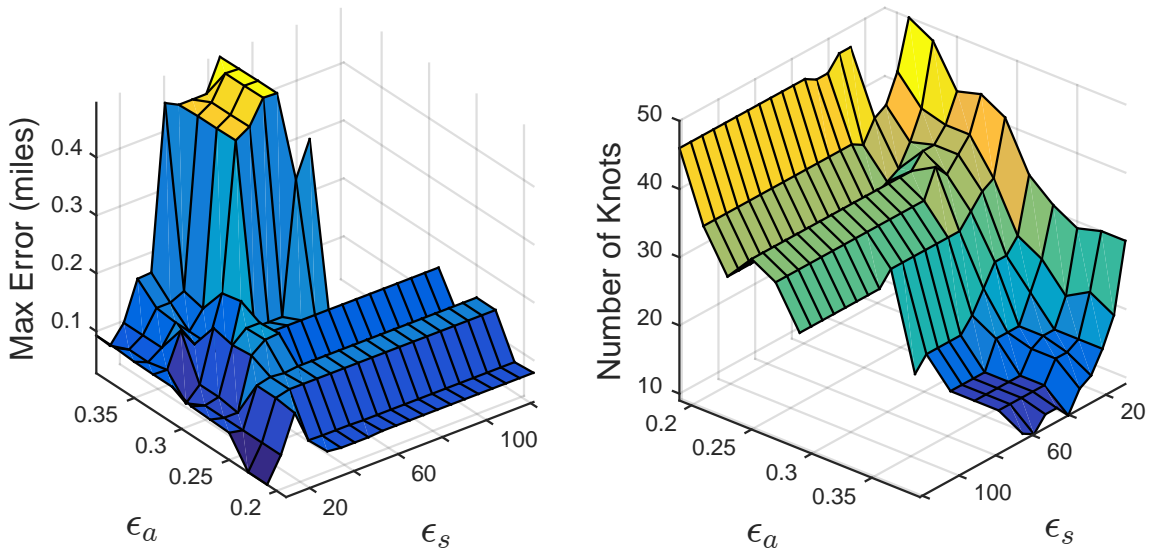


Figure 3. Spline Max Error and Number of Knots for y Trajectory

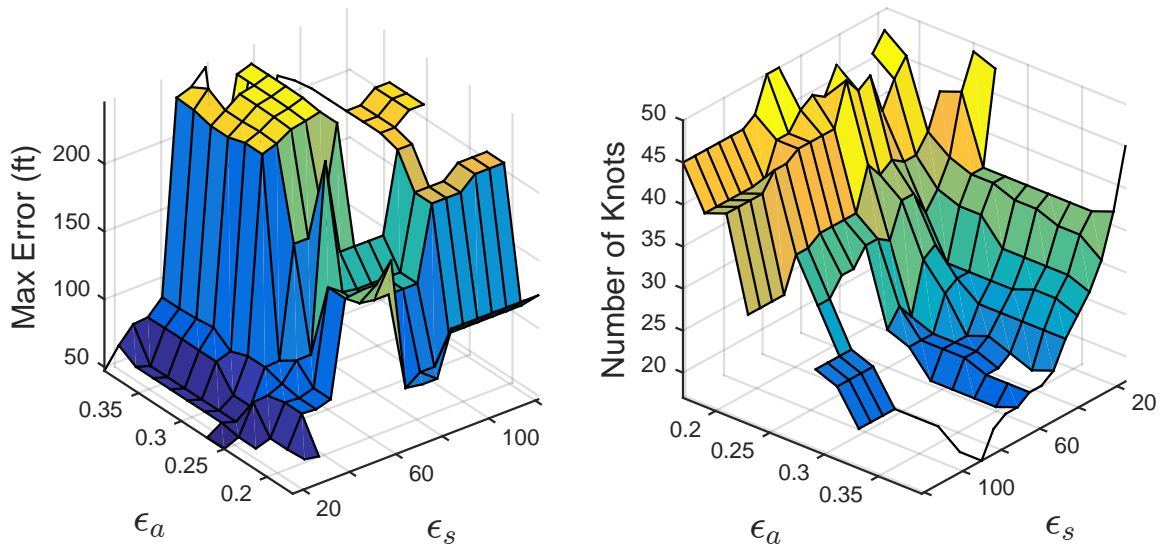


Figure 4. Spline Max Error and Number of Knots for z Trajectory

$$N_{comm} = 3(3(N + 1) + N)$$

for $N + 1$ sets of quadratic coefficients, N intervals between knots, and motion in three axes. In the worst case of 18 knots, this gives a total of 279 parameters; roughly half that many for Figure 6.

Figures 8, 9, and 10 display the splined altitude trajectories for the case where maximum error was held below 250 feet, 100 feet, and 50 feet, respectively. Again, the splines selected were the ones which matched the error criterion with the fewest knots, and the radar track is shown in blue and the spline fit is shown as a red dashed line. The knots are shown as green circles, and in these plots, the point of maximum error is highlighted with a black triangle. The additional (and from a trajectory planning perspective, unattractive) complexity of the altitude profile results in a requirement for more knots, but even in the worst case, 44 knots, that channel can be communicated with 50 foot accuracy with 176 numbers.

IV. Conclusions

This paper describes the first step in an effort to develop an optimization-based algorithm for cooperatively and efficiently arriving at runways on time, and generally operating in TRACON airspace without LoS. The first step has been to identify an attractive representation of the trajectories to be selected by optimization, and has been reported in this paper.

The optimization problem in which these trajectories will appear will be to simultaneously modify all planned trajectories in TRACON airspace to preserve LoS and appear at the runways when scheduled. The quadratic splines studied in this paper are conceptually attractive for this for several reasons:

- They are very simple, with low-order nonlinearity, which should encourage numerical docility in numerical optimization. Because the goal of the optimization will be to make small modifications to existing aircraft-generated trajectory plans to avoid LoS or schedule violation, we hope that the individual aircraft dynamics can be ignored.
- They can realize arbitrary motion without modifying their functional form. This is important in posing the optimization problem, and is contrasted with representations in which trajectories are pieced together logically via, say, straight lines and circular arcs.
- LoS is evaluated for continuous time via a closed form computation. This permits separation constraints to be posed as differentiable inequality constraints in the optimization problem.

The paper described the functional form of the quadratic splines proposed for the optimization, and the necessary conditions for computing their coefficients via least squares, and described a simple ad-hoc algorithm

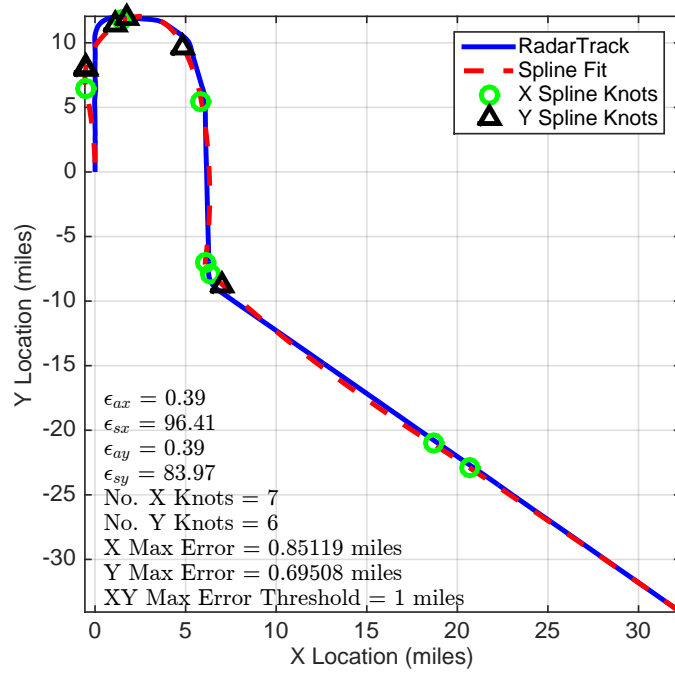


Figure 5. Splined Horizontal-Plane Trajectory: 1 Mile Error Threshold

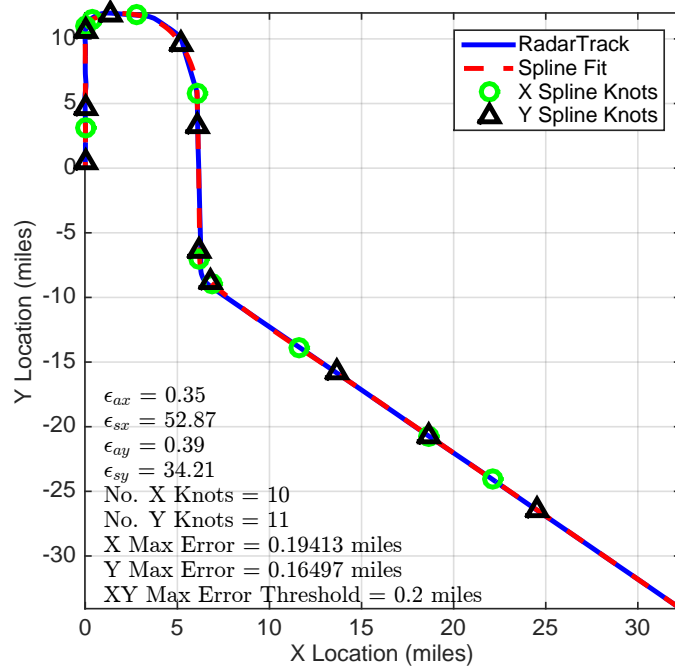


Figure 6. Splined Horizontal-Plane Trajectory: 0.2 Mile Error Threshold

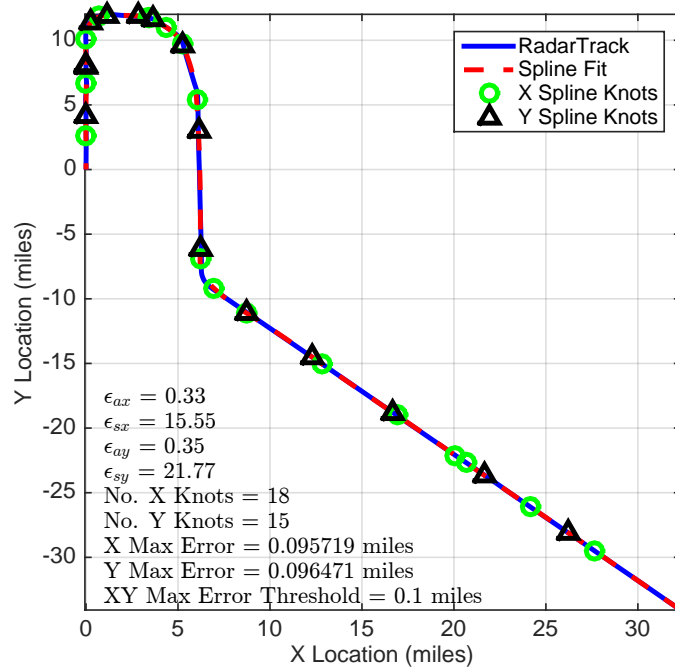


Figure 7. Splined Horizontal-Plane Trajectory: 0.1 Mile Error Threshold

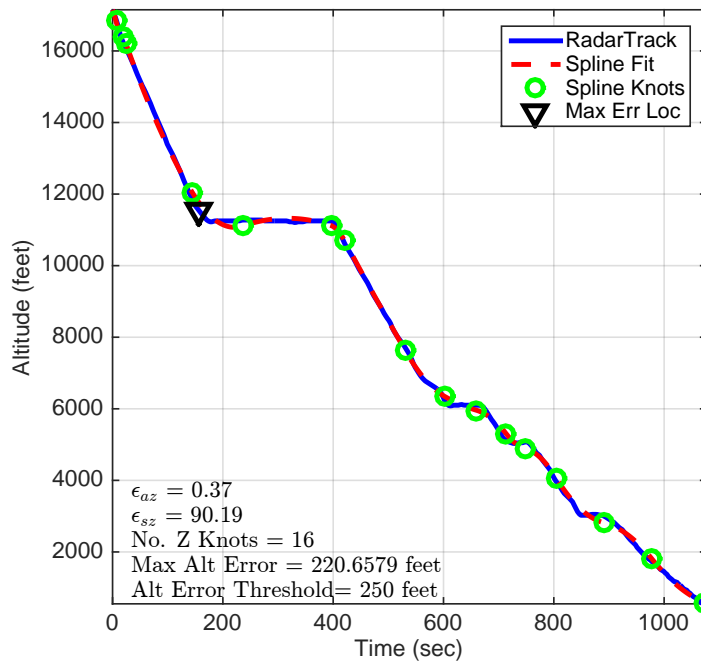


Figure 8. Splined Altitude Trajectory: 250 Foot Error Threshold

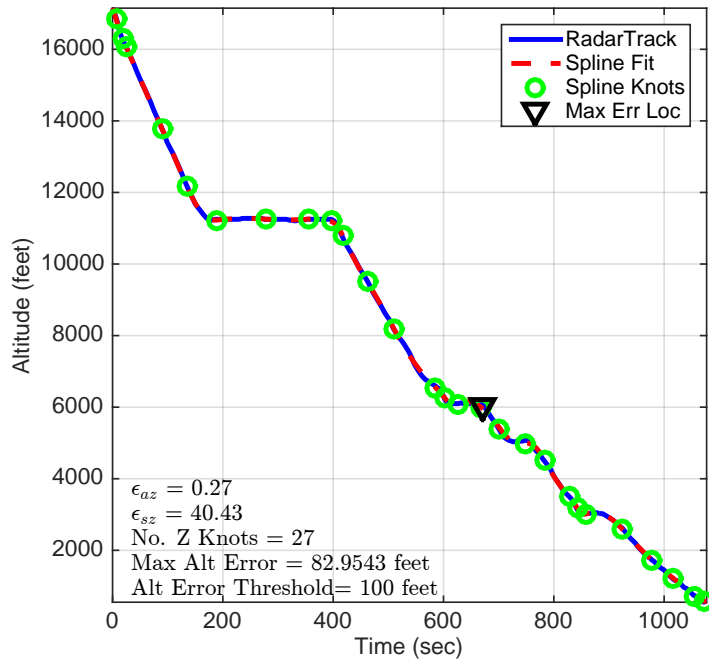


Figure 9. Splined Altitude Trajectory: 100 Foot Error Threshold

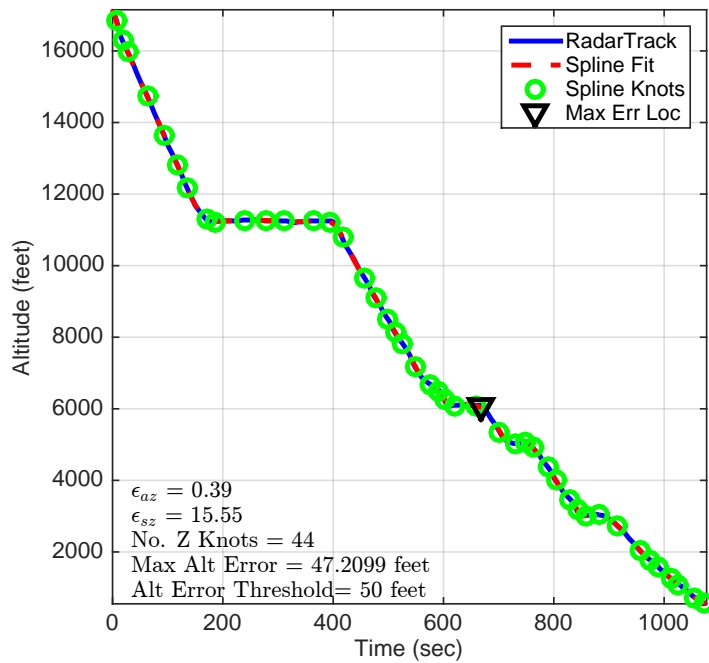


Figure 10. Splined Altitude Trajectory: 50 Foot Error Threshold

for placing knots. A numerical experiment evaluated the quality (max error) and cost (number of knots) of using quadratic splines to approximate some real-world TRACON trajectory data, though the types of trajectories appearing in the optimization process will almost certainly be less demanding.

References

¹Erzberger, H., and Itoh, E., "Design Principles and Algorithms for Air Traffic Arrival Scheduling," NASA TP-2014-218302.

²Slattery, R., and Zhao, Y., "Trajectory Synthesis for Air Traffic Automation," AIAA J. Guidance, Control, and Dynamics, Vol. 20, No. 2, March-April 1997, pp. 232-238.

³Wong, G., "The Dynamic Planner: The Sequencer, Scheduler, and Runway Allocator for Air Traffic Control Automation," NASA TM-2000-209586.

⁴Department of Transportation, Federal Aviation Administration, Air Traffic Control, Order JO 7110.65S, 14 February 2008.

⁵Ballin, M., and Erzberger, H., "An Analysis of Aircraft Landing Rates and Separations at Dallas/Ft. Worth Airport," NASA TM-1996-110397.