Mesh Refinement Using Density Function for Solving Optimal Control Problems

Yiming Zhao* and Panagiotis Tsiotras†

School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0150

This paper presents a general mesh refinement method based on a density function for distributing the grid points, which can be used for solving optimal control problems through a direct method. The algorithm captures any discontinuities and smoothness irregularities in the state and control variables with high resolution, thus helping with the convergence of the overall optimization algorithm. The algorithm also provides flexibility during the mesh refinement process, especially for problems with multiple control inputs. As a specific example of the general theory, a density function based on the best piecewise linear approximation of functions with curvature defined almost everywhere is proposed for refining the mesh. The technique is applied to solve the problem of the optimal power-limited landing for a large commercial aircraft.

I. Introduction

Numerical methods for solving optimal control problems generally fall into two general categories: indirect methods and direct methods.[1] The main idea of direct methods is to discretize the original continuous-time optimal control problem in a finite-dimensional nonlinear programming problem (NLP), and minimize *directly* the objective function of the original continuous-time optimal control problem. The solution of this NLP, which consists only of discrete variables, is used to approximate the continuous control and state histories. Based on the specific manner of time discretization of the original continuous-time problem, direct methods may be further classified into two major categories: direct shooting methods, and state and control parameterization methods.[2]

In direct shooting methods, only the control is parameterized, and explicit numerical integration (such as Runge-Kutta) is used to compute the state and control histories that satisfy the differential constraints (equivalently, the equations of motion), state and control constraints, as well as the problem boundary conditions. One advantage of direct shooting methods is the ability to formulate a problem in terms of a relatively small number of optimization variables.

The state and control parameterization methods involve a finite-dimensional approximation of both state and control variables using implicit Euler or Runge-Kutta schemes,[3] or collocation schemes.[4] These methods cause a considerable increase in the number of decision variables and equality constraints, which tends to slow down the optimization process. However, they tend to be numerically more robust, they are easy to implement, and can deal efficiently with state constraints. Collocation methods make the calculation of function gradients easier, hence sparse optimization techniques can be applied for higher efficiency.[5]

Although pseudospectral methods can also be classified as collocation methods, we list them here as a separate category because of their unique treatment of the discretization. Pseudospectral methods approximate the state and control histories using interpolating polynomials on an orthogonal finite basis.[2,4,6] They typically converge faster for smooth problems, exhibiting "spectral accuracy," but for problems with discontinuities, such as control switches or when hitting constraints, the smooth nature of the polynomials makes it difficult to capture accurately these discontinuities, resulting in a less accurate solution.

*Graduate Research Assistant, Email: yzhao7@gatech.edu.

†Professor, AIAA Fellow, Email: tsiotras@gatech.edu.

Much of the theory of numerical techniques for solving optimal control problems deals with proving the consistency between the solution of the discretized problem and that of the original continuous time problem. Conditions for the convergence of the solution of the discrete problem to that of the continuous problem have been developed in Refs. [7, 8]. In the same references, the convergence of second-order Runge-Kutta approximations to a nonlinear optimal control problem with control constraints was proved under the assumption that the optimal control has a derivative of bounded variation and that a coercivity condition holds for the cost function. A theoretical framework for consistent approximations dealing with the convergence of approximation problems was also introduced in Ref. [5].

For direct methods, a larger mesh size (number of points in the mesh) generally results in an improvement of the accuracy of the solution. A finer mesh also helps capture the discontinuities and smoothness irregularities in the controls and state variables more accurately, and hence helps achieve a better solution, albeit at the cost of more computation time. However, the accuracy of the solution depends not only on the mesh size, but also on the distribution pattern of the grid points. Even with the same mesh size, optimization results can be quite different when the corresponding meshes have different grid point distribution. Similar problems exist for the pseudospectral methods. Although pseudospectral methods do use non-uniform grids, nonetheless the grid pattern is fixed, regardless of the (non)smoothness of the solution. Hence, refinement strategies are needed to generate meshes to solve the discretized optimal control problems more accurately and efficiently.

There are two general methods for mesh refinement: static and dynamic.[5] In static refinements, after a solution of the discretized problem is computed, the same solution is used to refine the current mesh, by adding or moving points around using a certain strategy, so that the accuracy of the solution is improved in the next optimization step. In dynamic mesh refinements, some (or all) of the grid points are included as decision variables and allowed to move during optimization. Although dynamic mesh refinements may capture control discontinuities early on, they may also hinder convergence. They can also be less efficient than static mesh refinement strategies.[10] Recently, the issue of how to properly increase the level of discretization is receiving more attention: a mesh refinement technique for pseudospectral methods has been proposed, which partitions the whole time interval into smaller ones at the points of discontinuity, thus achieving a more accurate solution.[11] Reference [12] proposed a multi-resolution trajectory optimization algorithm (MTOA) for collocation schemes that automatically refines a nonuniform mesh after each iteration, so as to capture discontinuities and higher order smoothness irregularities of the control and state variables.[12]

In this paper, we improve on the work of Ref. [12] by proposing a general framework for mesh point allocation based on a grid point density function, which describes the pattern of the distribution of the grid points. The introduction of the mesh distribution function incorporates different mesh refinement schemes, and provides great flexibility to deal with problems involving multiple control inputs. A distribution function based on the best piecewise linear approximation is proposed and is incorporated into a density function based mesh refinement algorithm (DENMRA), which is subsequently applied to solve the optimal landing trajectory for a large commercial aircraft.

II. Density Function and Mesh Point Distribution

The choice of the mesh affects the quality of discretization and the convergence of the solution. A coarse uniform mesh may fail to capture important discontinuities and/or irregularities in the control and state variables, and may result in convergence to an inaccurate solution, or may even result in the failure of the overall optimization process. Improvements can be made by increasing the number of points in a uniform mesh for better resolution, but this results in the addition of unnecessary points at places without discontinuities or other irregularities in the solution. An efficient way to discretize an optimal control problem would be to use a non-uniform grid, which is dense only where it needs to be. In this section the concept of the grid point density function is introduced for generating meshes with specific patterns of distribution of their grid points.

II.A. Mesh Point Distribution on a Single Interval

A *density function* $\bar{f} : [0, 1] \to \mathbb{R}_+$ is a non-negative Lebesgue integrable function that satisfies $\int_0^1 \bar{f}(t)\, \mathrm{d}t = 1$. The corresponding *cumulative distribution function* $F : [0, 1] \to [0, 1]$ is defined by

$$F(t) \triangleq \int_0^t \bar{f}(\tau)\, \mathrm{d}\tau,$$

as shown in Fig.1. Clearly, $F(0) = 0$ and $F(1) = 1$. Any Lebesgue integrable, non-negative function $f : [0, 1] \to \mathbb{R}_+$ on the unit interval can be normalized to obtain a density function as follows:

$$\bar{f}\,(t) = \frac{f\,(t)}{\int_0^1 f\,(\tau)\, \mathrm{d}\tau}. \tag{1}$$

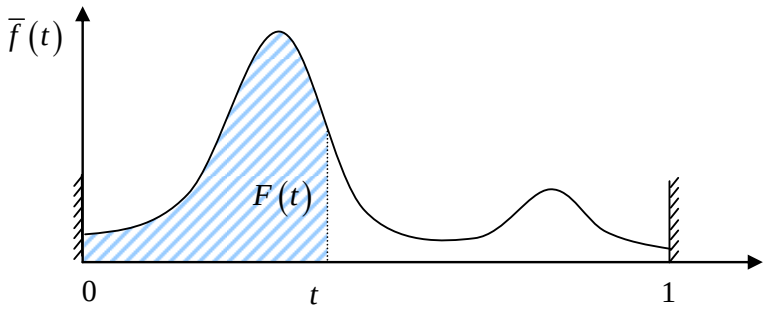


Figure 1. Density function and cumulative density function.

Consider a mesh $\{t_i\}_{i=1}^N$ containing a total of N points with $t_1 = 0$ and $t_N = 1$. Given the density function $\bar{f}$, let F be the cumulative distribution function determined by $\bar{f}$. For $i = 1, 2, \ldots, N-1$, with the i^{th} point at t_i, the position of the $(i+1)^{\text{th}}$ point can be decided by $F(t_{i+1}) - F(t_i) = 1/(N-1)$. This way, a mesh can be generated based on the density function $\bar{f}$, such that the distribution of grid points conforms to F, and the mesh is dense where the value of $\bar{f}(t)$ is large.

The previous mesh point allocation strategy usually requires solving a nonlinear algebraic equation repeatedly for $N-2$ times, which can be quite time-consuming when N is large. To reduce the computation time, instead of repeatedly solving the same nonlinear equation $N-2$ times, an interpolation method is used to compute the points $\{t_i\}_{i=2}^{N-1}$, by making use of the properties of F. Specifically, given any density function $\bar{f}$, select a partition $\{t_j\}_{j=1}^{N_j}$ of $[0,1]$, which can be either a uniform grid with $N_j \geq N$ points for an accurate solution of the grid $\{t_i\}_{i=2}^{N-1}$, or a nonuniform grid generated in the previous step in an iterative algorithm. Now $y_j = F(t_j)$ can be easily calculated by $y_j = \int_0^{t_j} \bar{f}(\tau)\, \mathrm{d}\tau$. Because F is monotone and continuous on $[0,1]$, its inverse mapping F^{-1} exists and is also continuous, with $t_j = F^{-1}(y_j)$. Then $\{(y_j, t_j)\}_{j=1}^{N_j}$ is a discrete representation of the function F^{-1}. It is known that the first and the last grid points are at $t_0 = 0$ and $t_N = 1$, respectively. For the allocation of other grid points, $F^{-1}(t)$ is approximated by a spline function such that the position of i^{th} mesh point t_i can be obtained by interpolating $\{(y_j, t_j)\}_{j=1}^{N_j}$ using the spline function at $y_i = (i-1)/(N-1)$ for all $2 \leq i \leq N-1$. Using this method, the location of all mesh points can be calculated very fast and with high accuracy.

As an example, Fig. 2 shows the mesh point distribution determined by two specific density functions. The raw density function in the upper left of the figure is the linear function $f(t) = t$. The resulting mesh is shown in the upper right of the figure. The lower left shows the raw density function $f(t) = e^{-50t^2+20t-2} + e^{-50t^2+80t-32}$, with its mesh shown in the lower right of figure. In both cases, the mesh contains a total of 20 grid points.

II.B. Mesh Point Distribution on Disjoint Intervals

When increasing the size of the mesh, it may be required that new points are added only within certain specific spans of the control and state histories, namely at places where the control or state histories exhibit

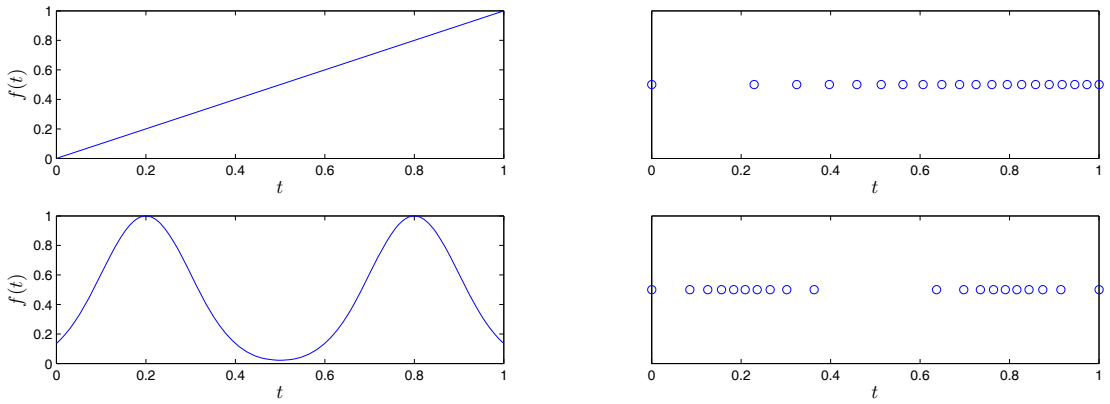


Figure 2. Density functions and corresponding meshes.

discontinuities or smoothness irregularities (e.g., very fast rate of change and/or discontinuities in higher order derivatives) while keeping other points fixed. To provide this flexibility, the method introduced in the former section is extended to support mesh point distribution on disjoint intervals.

To this end, suppose that a total of N new points need to be added to S disjoint intervals $\mathcal{I}_s \subset \mathbb{R}_+,\ s = 1, \ldots, S$, where $\mathcal{I}_s = [t_{sa}, t_{sb}],\ 1 \leq s \leq S$, and $t_{ib} < t_{ja}$ for all $1 \leq i < j \leq S$. The distribution of grid points is described by S density functions $f_s : \mathcal{I}_s \to \mathbb{R}_+,\ s = 1, 2, \ldots, S$ defined on the corresponding intervals. With a total of $2S$ points located at both ends of the S intervals, the remaining $N - 2S$ points should be allocated in the interior of the intervals conforming to certain appropriately chosen density functions. This can be done by assigning to each interval an appropriate number of points based on the density functions on all the intervals. Specifically, the number of points assigned to the s^{th} interval $\mathcal{I}_s$ can be determined by the formula

$$N_s = \text{round}\left((N - 2S) \frac{\int_{t_{sa}}^{t_{sb}} f_s(t)\mathrm{d}t}{\sum_{p=1}^{S} \int_{t_{pa}}^{t_{pb}} f_p(t)\mathrm{d}t} \right) + 2,$$

where round$(\cdot)$ denotes rounding to the nearest integer. Note that there may be one point leftover after the mesh point allocation for all intervals. If this happens, the last point can be assigned to the interval with the greatest rounding error. In the extreme case of equal rounding errors, the remaining point can be assigned arbitrarily to any interval.

After all points have been allocated, the meshes can be generated on each interval $\mathcal{I}_s$ using the method described in the previous section. To this end, we first transform the interval $\mathcal{I}_s$ to $\mathcal{I} = [0, 1]$ using a function $\psi_s : \mathcal{I}_s \to \mathcal{I}$ defined by $\psi_s = K_s \circ D_s$, where $D_s : \mathbb{R} \to \mathbb{R}$ is a displacement function given by $D_s(t) = t - t_{sa}$, and K_s is a scalar given by $K_s = 1/(t_{sb} - t_{sa})$. The inverse transformation ψ_s^{-1} is also well defined, and the corresponding density functions $\bar{f}_s$ are given by the normalization of $f_s(\psi_s^{-1}\tau)$, which are used for mesh generation. Since ψ_s is a homeomorphism between $\mathcal{I}_s$ and $\mathcal{I}$, the mesh T_s' generated on $\mathcal{I}$ can be transformed back to T_s on $\mathcal{I}_s$ using the inverse transformation ψ_s^{-1}. The final mesh on the disjoint intervals is given by $T = \bigcup_{s=1}^{S} T_s$. The idea is illustrated in Fig. 3.

III. Density Function Mesh Refinement Algorithm (DENMRA)

Most optimization algorithms compute stationary points, which may not approximate well the solution of the original optimal control problem. For better precision, a finer discretization becomes progressively crucial as the iteration gets closer to the optimal solution,[9] especially when the controls and/or their derivatives exhibit discontinuities or smoothness irregularities. The solution of the optimization may achieve a better approximation with a properly refined discretization.

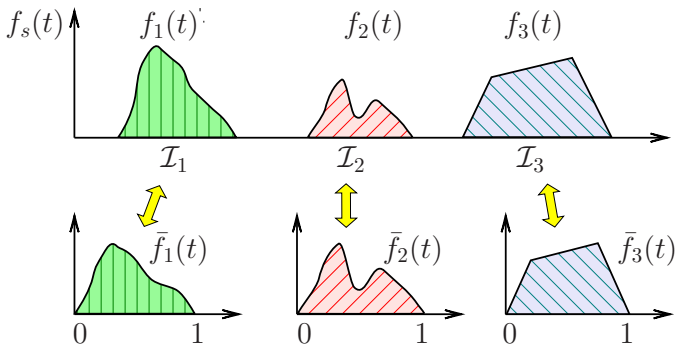


Figure 3. Joint density distribution on disjoint intervals.

The DENsity function based Mesh Refinement Algorithm (DENMRA) is an iterative algorithm for solving optimal control problems. It uses a static mesh refinement strategy that updates the current mesh using a density function obtained from the previous iteration. This density function, as shown in the Appendix, provides the best piecewise linear approximation (in the $\mathcal{L}^1$-norm) of the function of interest using C^2 functions. General optimal control problems involve ordinary differential equations in terms of the state variables, which describe how the controls changes the vector field of the states. For such problems, since the controls are bounded, the states are always continuous, and any smoothness irregularities in the states correspond to some irregular changes in control. Hence it is enough to consider only the controls from the previous iteration to capture the smoothness irregularities in both the states and the controls.

The mesh generated using the density function ρ_Γ (see the Appendix) is characterized by an appropriate local level of discretization over the whole control history, which helps capture the discontinuities or smoothness irregularities during the optimization process. More details about the density function ρ_Γ and its properties can be found in the Appendix.

III.A. Major Steps of DENMRA

When solving a general optimal control problem that minimizes the cost function J using m control inputs, the DENMRA goes through the following four major steps:

(1) Set $j = 1$. Choose a positive integer N_j, and solve the discretized problem that minimizes J using a uniform mesh $T_1 = \{t_i\}_{t_i=1}^{N_j}$, where $t_i = (i-1)/(N_j - 1)$;

(2) Calculate the density function $\bar{f}$ using the discretized control $\{(t_i, \mathbf{u_i})\}_{i=1}^{N_j}$ of the solution, where $\mathbf{u_i} \in \mathbb{R}^m$;

(3) Let $N_{j+1} = N_j + \Delta N_j$ for some non-negative integer ΔN_j, which is an increment of the mesh size, and generate the new mesh $T_{j+1} = \{t_i\}_{i=1}^{N_{j+1}}$ based on $\bar{f}$. Set $j = j + 1$;

(4) Solve the problem discretized with mesh T_j, and go to Step (2), unless some stopping rule is met.

The details of the DENMRA such as the subroutines and the setting of parameters are introduced in the immediate section.

III.B. Details of DENMRA

III.B.1. Initial Guess

For simplicity, the DENMRA may start from a constant initial guess for all control and state variables.

III.B.2. Optimization

The DENMRA calls SNOPT[13] for the optimization of the discretized optimal control problem. SNOPT is a sparse optimization software, which solves efficiently the NLP using collocation.

III.B.3. Density Function Computation

As stated in the Appendix, for any C^2 a.e. function Γ, the density function ρ_Γ which yields the best piecewise linear approximation of Γ is given by $\rho_\Gamma(t) = c\kappa(t)^{1/3}$, $t \in [0,1]$, where c is a constant and κ is the curvature of Γ, given by

$$\kappa = \frac{|\Gamma''|}{|(1+\Gamma'^2)^{3/2}|}. \tag{2}$$

Because Γ is not known, κ is evaluated using the discrete control values $\{(t_i, \mathbf{u_i})\}_{i=1}^{N_j}$ from the previous iteration. The calculation of the density function $\bar{f}$ corresponding to the controls is as follows:

(1) Let $u_{i,k}$ be the k^{th} component of the discrete control value $\mathbf{u_i}$, $\dot{u}_{i,k}$ be the first order derivative of the k^{th} component of control at time $t'_i = (t_{i+1} + t_i)/2$, and $\ddot{u}_{i,k}$ be the second order derivative at time $t''_i = (t'_{i+1} + t'_i)/2$. Then, for $k = 1, \ldots, m$, the values $\{\dot{u}_{i,k}\}_{i=1}^{N_j-1}$ and $\{\ddot{u}_{i,k}\}_{i=1}^{N_j-2}$ can be approximated by $\dot{u}_{i,k} \approx (u_{i+1,k} - u_{i,k})/(t_{i+1} - t_i)$ and $\ddot{u}_{i,k} \approx (\dot{u}_{i+1,k} - \dot{u}_{i,k})/(t'_{i+1} - t'_i)$, respectively. Interpolate $\{(t'_i, \dot{u}_{i,k})\}_{i=1}^{N_j-1}$ using a spline function at t''_i and obtain $\{(t''_i, \dot{u}'_{i,k})\}_{i=1}^{N_j-2}$.

(2) Let

$$\kappa_{i,k} = \frac{|\ddot{u}_{i,k}|}{|(1+\dot{u}'^2_{i,k})^{3/2}|}, \qquad i = 1, \ldots, N_j - 2. \tag{3}$$

The curvature function $\kappa_k(t)$ for the k^{th} control component is constructed using interpolation and/or extrapolation of $\{(t''_i, \kappa_{i,k})\}_{i=1}^{N_j-2}$ using a spline function. Then $\rho_{\Gamma_k}(t) = c_k\kappa_k(t)^{\frac{1}{3}}$, as shown in the Appendix.

(3) The overall non-normalized density function f is obtained by merging the density functions for all controls. For instance,

$$f(t) = \left(\sum_{k=1}^{m} \rho_k^2(t)\right)^{\frac{1}{2}}, \tag{4}$$

and

$$f(t) = \max_k \rho_k(t) \tag{5}$$

are two methods to generate the overall non-normalized density function.

When the control histories are straight lines on some interval $\mathcal{I}_s \subset \mathbb{R}$, f is constant on $\mathcal{I}_s$, and theoretically no points need to be allocated to $\mathcal{I}_s$, except for its two endpoints. However, as the control history on $\mathcal{I}_s$ may change in subsequent iterations, it is a good idea to keep some points in the interior of the interval $\mathcal{I}_s$ in order to capture possible changes of the control histories. For this reason f is modified as follows to keep some points on the parts of the control histories that are straight lines or curves with very small curvatures:

$$f_\varepsilon(t) = \begin{cases} f(t), & f(t) \geq \varepsilon, \\ \varepsilon, & f(t) < \varepsilon, \end{cases}$$

where ε is a small positive real number. Finally, f_ε is normalized as in equation (1) to get $\bar{f}$.

III.B.4. Mesh Generation

The method for generating a mesh with N points based on a density function $\bar{f}$ has been already described in detail in Section II. Another issue with the mesh generation step is how to determine the size of the mesh.

Although the DENMRA can redistribute the current grid points and obtain a refined mesh while keeping the mesh size constant, increasing the mesh size makes the mesh even finer and helps improve the accuracy of the solution. On the other hand, the trade-off between precision improvement and a longer optimization time as a result of a larger mesh size, also needs to be taken into account.

The DENMRA typically starts with a coarse uniform mesh (a mesh of about 20 points usually suffices) in order to capture the basic structure of the control history. The final mesh, typically after 3 to 5 iterations, may include twice as many points. The user can adjust the final mesh size and the number of iterations for different speed and accuracy requirements when dealing with different problems. The mesh size is increased by roughly the same number of points ΔN_j after each iteration, such that the final mesh size is as specified. Specifically,

$$\Delta N_j = \text{round}\left(\frac{N_{j_{\max}} - N_j}{j_{\max} - j}\right), \qquad i = 1, \ldots, j_{\max} - 1,$$

where $j_{\max}$ is the total number of iterations, and $N_{j_{\max}}$ is the size of the final mesh.

III.B.5. Stopping Rule

Currently, the stopping criterion of the DENMRA is to simply specify the maximum number of iterations. It is of course possible to implement other stopping rules using the intermediate results of the algorithm, such as the norm of the difference of the controls between two subsequent iterations.

IV. Comparison of DENMRA with Uniform and MOTA Discretizations

A double integrator minimum energy problem[14] is used to test the performance of the DENMRA against a uniform discretization and the non-uniform discretization generated by the MOTA algorithm proposed in Ref. [12]. The problem is given by:

$$\begin{aligned} \dot{v} &= u, \qquad & v(0) &= -v(1) = 1, \\ \dot{x} &= v, \qquad & x(0) &= x(1) = 0, \end{aligned}$$

and the goal is to find $u(t)$, where $0 \le t \le 1$, to minimize

$$J = \frac{1}{2}\int_0^1 u^2 \, \mathrm{d}t,$$

with the constraint $x(t) \le \ell$, where ℓ is a positive real number.

The theoretical solution of the optimal control $u^*(t)$ can be obtained as follows:[14]

$$u^*(t) = -2, \;\; 0 \le t \le 1, \qquad \text{for } \ell \ge \tfrac{1}{4};$$

$$u^*(t) = \begin{cases} -8(1-3\ell) + 24(1-4\ell)t, & 0 \le t \le \frac{1}{2}, \\ -8(1-3\ell) + 24(1-4\ell)(1-t), & \frac{1}{2} < t \le 1, \end{cases} \quad \text{for } \tfrac{1}{6} \le \ell < \tfrac{1}{4};$$

$$u^*(t) = \begin{cases} -\frac{2}{3\ell}(1 - \frac{t}{3\ell}), & 0 \le t \le 3\ell, \\ 0, & 3\ell < t \le 1 - 3\ell, \\ -\frac{2}{3\ell}(1 - \frac{1-t}{3\ell}), & 1 - 3\ell < t \le 1, \end{cases} \quad \text{for } \ell < \tfrac{1}{6}.$$

IV.A. Comparison of Precision and Speed

Table 1 compares the precision and speed of the different algorithms used to solve the double integrator problem. In Table 1 N is the size of the final mesh, $|J - J^*|$ is the optimality error, and $\|u_i - u^*(t_i)\|_\infty = \max_i |u_i - u^*(t_i)|$ is the infinity norm of the error between the discretized control $\{u_i\}_{i=1}^N$ and the exact solution u^*.

The MTOA starts with 17 points, and the DENMRA starts with 20 points. For the MTOA, the number of the grid points N is updated by the algorithm, and cannot be determined a priori, while for the DENMRA N is selected such that the algorithm achieves the same order of accuracy as the MTOA, for comparison. The value of N is constant for both Uniform 1 and Uniform 2 cases. All algorithms were tested on the same computer with 2.40GHz Intel Core2 Quad CPU and 3G RAM, and cold-started from the same linear initial guess.

Table 1. Comparison of precision and speed.

ℓ	Algorithm	N	$\lvert J - J^* \rvert$	$\lVert u_i - u^*(t_i) \rVert_\infty$	CPU time (s)
0.04	Uniform 1	50	2.9e-1	8.8e-1	0.4
	Uniform 2	200	1.9e-2	3.1e-1	2.6
	MTOA	49	5.2e-6	8.6e-4	3.6
	DENMRA	30	4.5e-7	3.9e-4	1.9
0.08	Uniform 1	50	3.8e-2	2.7e-1	0.3
	Uniform 2	200	2.3e-3	8.2e-2	2.4
	MTOA	55	2.7e-7	2.1e-4	3.6
	DENMRA	30	6.4e-9	2.9e-4	1.9
0.12	Uniform 1	50	1.1e-2	1.1e-1	0.3
	Uniform 2	200	7.1e-4	3.7e-2	2.3
	MTOA	45	4.1e-8	3.5e-5	3.1
	DENMRA	30	1.8e-8	3.8e-5	2.0
0.16	Uniform 1	50	5.0e-3	7.8e-2	0.3
	Uniform 2	200	3.0e-4	2.1e-2	2.2
	MTOA	61	2.5e-8	2.1e-4	3.3
	DENMRA	30	8.2e-9	1.7e-4	2.1

For this problem, the precision of the uniform mesh discretization schemes is not satisfactory. Optimality and precision of the solution can only be slightly improved by increasing N from 50 to 200, and the optimization fails when $N = 300$. On the other hand, both the MTOA and the DENMRA exhibit much better precision over the uniform mesh scheme, with optimality error of order $10^{-7} \sim 10^{-9}$, and maximum control error of order $10^{-4} \sim 10^{-5}$.

The MTOA is based on dyadic grids and it starts the mesh refinement from a baseline mesh consisting of $2^i + 1$ points, for $i = 1, 2, \ldots$ (usually we take $i = 4$ to keep the computation time short while capturing the basic structure of the control history). The mesh generated by the MTOA is nonuniform with a dyadic structure. This fixed dyadic structure reduces the flexibility for re-allocating grid points arbitrarily during the mesh refinement process. The DENMRA, on the other hand, generates completely nonuniform and unstructured meshes.

While increasing the mesh size usually results in higher precision, the DENMRA can also obtain a more precise solution by simply redistributing the grid points while keeping the same mesh size. This unique characteristic of the DENMRA is illustrated in Figs. 4 and 5, which record the mesh refinement process of the DENMRA when solving the double integrator problem with $\ell = 0.09$. The theoretical solution of the optimal control u^* is shown in Fig. 6. The mesh size is fixed at $N = 20$, and the DENMRA improves the precession of the solution by only redistributing these 20 grid points. With a slight abuse of notation, here f_j denotes the density function after the j^{th} iteration. It is obvious from Fig. 5 that after each iteration, the grid points get denser around the two points of discontinuity, thus providing a better resolution. For this specific example, the DENMRA found the solution in 1.33 sec, with an optimality error of $|J - J^*| = 1.3 \times 10^{-6}$ and a maximum control error of $\|u_j - u^*(t_j)\|_\infty = 3.33 \times 10^{-4}$.

For the double integrator problem, on average, the DENMRA saves 42% of the computation time compared to the MTOA for the test cases listed in Table 1.

Figure 7 compares the convergence speed of the MTOA and the DENMRA for the double integrator

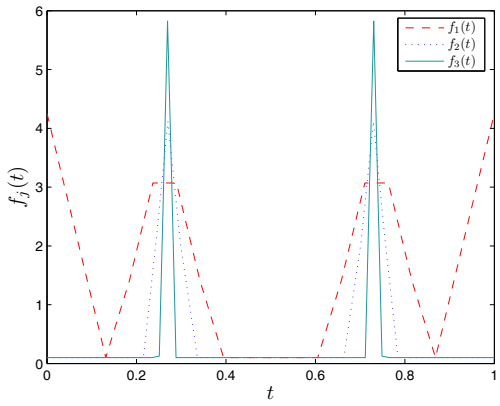


Figure 4. Density functions after each iterations.

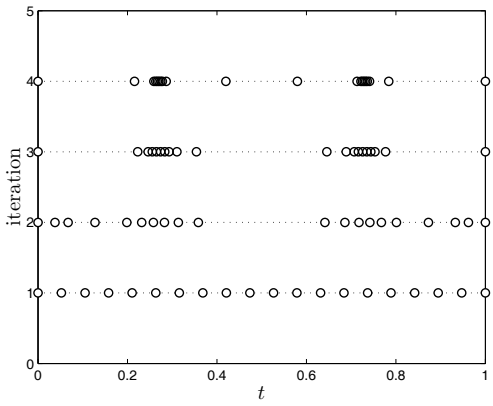


Figure 5. Mesh refinement history.

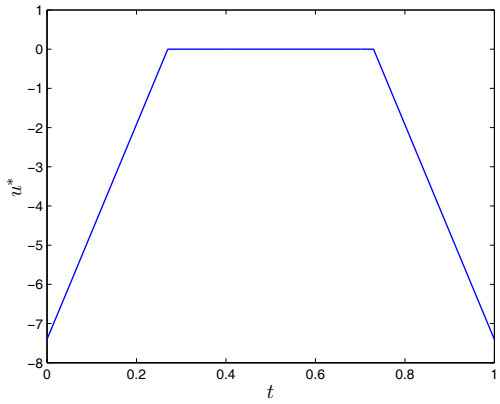


Figure 6. Optimal control for $\ell = 0.09$.

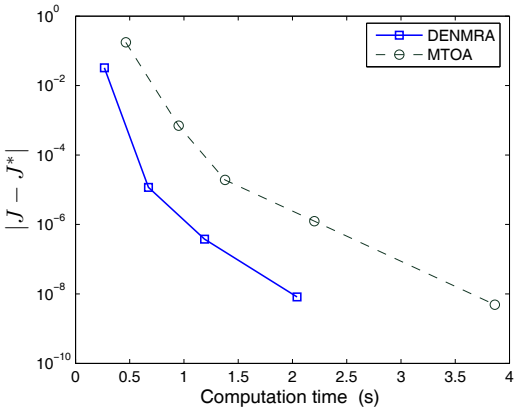


Figure 7. Convergence Speed for $\ell = 0.1$.

problem when $\ell = 0.1$. It is clear from Fig. 7 that the DENMRA achieves the same order of accuracy in less time, or equivalently, achieves higher accuracy within the same time.

IV.B. Comparison of Resolution

By "resolution" here we mean not only the ability of an algorithm to capture the discontinuities in the control history, but also the ability to distinguish adjacent points of discontinuity. The theoretical solution of x for the double integrator problem is shown in Fig. 8.

(1) When $\ell \geq 1/4$, $u^*(t)$ is constant, and all algorithms converge nicely to the theoretical solution.

(2) When $1/6 \leq \ell < 1/4$, the discontinuity of the control occurs at $t = 1/2$, which happens to be exactly the middle point of the starting mesh of MTOA, thus MTOA achieves a good interpolation accuracy at the first iteration, and fails to continue refining the mesh after the first iteration, since the interpolation error is lower than the threshold set priori in the algorithm.The DENMRA can still re-distribute the points after the first iteration, and thus achieves higher accuracy.

(3) When $\ell < 1/6$, the optimal control $u^*(t)$ exhibits two corners, and the problem becomes very challenging when ℓ tends to zero or $1/6$: in the former case, the discontinuities are very close to the two endpoints of the mesh, where the precision of integration and interpolation methods used deteriorates;

in the second case, the two points of discontinuity tend to merge, which also makes them difficult to separate.

Table 2. Comparison of resolution.

Algorithm	ℓ	Description ID	Resolution (Δt)	$\lvert J - J^* \rvert$	$\lVert u_i - u^*(t_i) \rVert_\infty$
MTOA	0.038	D1	0.114	5.0e-6	1.0e-3
	0.1640	D2	1.6e-2	1.1e-8	5.6e-4
DENMRA	0.008	D3	0.024	5.6e-6	5.6e-3
	0.1662	D4	2.8e-3	1.6e-9	5.8e-4
D1: the smallest ℓ without algorithm failure D2: the largest ℓ with which the discontinuities are still separated D3: the smallest ℓ which keeps $\lVert u_i - u^*(t_i) \rVert_\infty < 10^{-2}$ D4: the largest ℓ with which the discontinuities are still separated					

Table 2 compares the resolution of the MTOA and the DENMRA. Here the resolution is denoted by Δt. When $\ell \to 0$, $\Delta t = 3\ell$, where Δt is the distance between the discontinuities and the nearby endpoints of the mesh. When $\ell \to 1/6$, $\Delta t = 1 - 6\ell$, which is the distance between the two points of discontinuity. In both cases, a smaller Δt means a better resolution. For all test cases the DENMRA starts from a uniform mesh of 20 points and terminates with 30 points, while the MTOA starts from 17 points and terminates with approximately 60 points. As shown in Table 2, the DENMRA provides sharper resolution than the MTOA, while keeping the high accuracy of the solution.

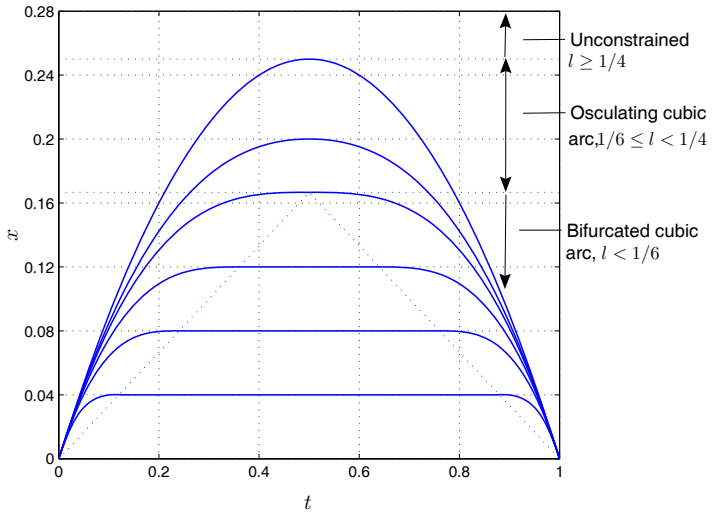


Figure 8. Theoretical solution of $x(t)$ for several values of ℓ.

V. Optimal 3D Landing Trajectory for a Large Civilian Aircraft

In order to test the robustness of the DENMRA for solving more complicated optimal control problems, several optimal landing scenarios were studied for the DC9-30 aircraft. The kinematic and dynamical

equations of the aircraft are as follows:

$$\dot{x} = \upsilon \cos\gamma \cos\psi \tag{6}$$
$$\dot{y} = \upsilon \cos\gamma \sin\psi \tag{7}$$
$$\dot{z} = \upsilon \sin\gamma \tag{8}$$
$$\dot{\upsilon} = \frac{1}{m}\left(T\cos\alpha - D(\alpha, v, z)\right) - g\sin\gamma \tag{9}$$
$$\dot{\gamma} = \frac{1}{m\upsilon}(T\sin\alpha + L(\alpha, v, z)\cos\phi) - \frac{g}{\upsilon}\cos\gamma \tag{10}$$
$$\dot{\psi} = -\frac{1}{m\upsilon\cos\gamma}L(\alpha, v, z)\sin\phi \tag{11}$$

where x, y are the position of the aircraft in the horizontal plane, z is the altitude, υ the airspeed, γ the flight path angle, g the gravitational acceleration, ψ the heading angle, α the angle of attack, and ϕ the bank angle. T is the thrust, $L(\alpha, v)$ the lift force and $D(\alpha, v)$ is the drag force acting on the aircraft. In this model, the control variables are the angle of attack α, the bank angle ϕ and the thrust T. In a commonly accepted approximation, the lift and drag forces are functions of α and v, as described in following equations:

$$D(\alpha, v, z) = Q(v, z)SC_D(\alpha),$$
$$L(\alpha, v, z) = Q(v, z)SC_L(\alpha),$$

where $Q(v, z)$ is the dynamic pressure given by $Q(v, z) = \frac{1}{2}\rho(z)v^2$, $\rho(z)$ the air density at altitude z, and S is the wing surface area. The lift and drag coefficients $C_L(\alpha)$ and $C_D(\alpha)$ can be calculated, as usual, by the following equations,

$$C_L(\alpha) = C_{L_0} + C_{L_\alpha}\alpha,$$
$$C_D(\alpha) = C_{D_0} + KC_L^2(\alpha),$$

where C_{L_0} is the lift coefficient at zero angle of attack, and C_{L_α} is the lift coefficient slope. C_{D_0} accounts for the drag of the whole aircraft, and the second term in $C_D(\alpha)$ accounts for the induced drag, specifically, $K = 1/(0.95e\pi \quad)$, where e is the efficiency factor, which is corrected by 0.95 for the assumed landing configuration. is the aspect ratio of the aircraft defined by $= b^2/S$, where b is the wing span. In the current model, it is assumed that the mass of the aircraft m is constant. Since large civil aircraft usually fly at very high altitude, an accurate atmospheric model is used for solving the optimal landing problem.[16] The values of the parameters in the former equations are given in Table 3, where $T_{\max}$ is the maximum engine thrust.

Table 3. Parameters for DC9-30.

m	g	ρ	S	C_{L_α}	C_{L_0}	C_{D_0}	K	$T_{\max}$
49,940 kg	9.8 kgm/s^2	1.225 kg/m^3	112 m^2	4.2	0.4225	0.0197	0.0459	137.81kN

V.A. Initial Guess for NLP Optimization

Any NLP solver needs an initial guess of the decision variables to start the iteration process. For minimum-time optimal control problems, the decision variables include discretization of states, controls and final time. Because the problems of interest are generally not convex, it is important to provide a reasonable initial guess for the solver to converge. If the problem is well behaved, simple affine or constant initial guesses will typically be enough, otherwise, the user will need to provide a reasonable initial guess, which is often difficult.

Finding a good initial guess turns out to be challenging for large civil aircraft landing trajectory optimization. Large civil aircraft usually cruise at an altitude of around $10,000$ m, where the air density is about 0.4140kg/m^3, which is only 33.8% of the value at the sea level. Constantly changing air density during the landing process makes it more difficult for the NLP solver to converge, especially when the initial guess is

not good. Numerical experiments have shown that an arbitrary affine or constant initial guess of states and controls works satisfactorily for the constant air density scenario, but it is difficult to find a converging initial guess for the real air density scenario. Experience and a good intuition about the shape of the optimal trajectory in (x, y, z) space can help, but not so for the velocity profile and time histories of the required controls to fly such a trajectory. Besides, if the initial guess of states and controls are not dynamically consistent, the solver may also fail to converge for a sensitive problem.

In this paper, for the sensitive cases in which the landing problem with real air density and constant initial guess fails to converge, the same problem with constant air density is solved, and the result is used as an initial guess for solving the real air density problem again. For all sensitive cases which have been tested, this procedure led to convergent solutions.

V.B. Zero-Thrust Landing in Minimum Time

When an aircraft loses thrust because of engine failure, fuel depletion, or any other unforeseeable problems, a reasonable option to guarantee the safety of the passengers is to land the aircraft at a nearby airport as soon as possible. This can be treated as a minimum-time optimal control problem with fixed boundary conditions. In this section we consider two cases for the zero-thrust, minimum-time landing problem. For both cases the aircraft loses power at an altitude $z = 10\,\text{km}$, cruise speed $v = 240\,\text{m/s}$, and flight path angle $\gamma = 0°$, and needs to land at a nearby airport using only the angle of attack α and the bank angle ϕ as control inputs. In the first case, the position of the airport is at $(60, 50, 0)\,\text{km}$, with runway direction $\psi = -10°$. In the second case, the position of the airport is at $(120, -30, 0)\,\text{km}$, with runway direction $\psi = 30°$.

In the first case, the initial conditions for the states and controls are $x(t_0) = 0\,\text{km}$, $y(t_0) = 0\,\text{km}$, $z(t_0) = 10\,\text{km}$, $v(t_0) = 240\,\text{m/s}$, $\gamma(t_0) = 0°$, $\psi(t_0) = 0°$, $\alpha(t_0)$ free, $\phi(t_0) = 0°$. The final conditions are $x(t_f) = 60\,\text{km}$, $y(t_f) = 50\,\text{km}$, $z(t_f) = 0\,\text{km}$, $v(t_f) = 95\,\text{m/s}$, $\gamma(t_f) = -3°$, $\psi(t_f) = -10°$, $\alpha(t_f)$ is free, and $\phi(t_f) = 0°$. Both $\phi(t_0)$ and $\phi(t_f)$ are set to zero for consistency with the normal attitude of an aircraft during the level flight and touch-down phases. For the second case, only some of the final conditions are changed, that is, $x(t_f) = 120\,\text{km}$, $y(t_f) = -30\,\text{km}$, and $\psi(t_f) = 30°$. The same state and control constraints were used for both cases, namely, $\alpha_{\min} = -5°$, $\alpha_{\max} = 15°$, $\phi_{\min} = -15°$, $\phi_{\max} = 15°$, $v_{\min} = 75\,\text{m/s}$, $v_{\max} = 260\,\text{m/s}$, $\gamma_{\min} = -7°$ and $\gamma_{\max} = 10°$.

The cost function is given by $J = J_1 + J_2$, where

$$J_1 = t_f,$$

which is the total flight time (the main goal of optimization), and J_2 is an additional term that helps with the numerical calculations.

Spurious oscillations (wiggles) of the variables often arise in numerical schemes. These can be characterized by a large total variation of the first time derivative. Hence, it is possible to reduce the oscillations of the state and control variables by penalizing the weighted summation of the total variations (STV) of the corresponding first time derivatives $\dot{x}_j$ and $\dot{u}_i$. The total variation is usually defined using the $\mathcal{L}_1$-norm, but here we measure oscillation using the $\mathcal{L}_2$-norm for convenience;

$$J_2 = \left(\int_{t_0}^{t_f} \left(\sum \mu_i \ddot{u}_i^2 + \sum \lambda_j \ddot{x}_j^2 \right) \mathrm{d}t \right)^{\frac{1}{2}}.$$

Specifically, J_2 is the weighted $\mathcal{L}_2$-norm of the second order time derivatives $\ddot{x}_j$ and $\ddot{u}_i$, and is used to reinforce smoothness of the solution. J_2 is similar to STV in terms of measuring excessive oscillations, but the quadratic form of the integrand enhances convergence. The weights μ_i and λ_j are penalty coefficients which can be adjusted to achieve satisfactory smoothness for all the variables. Under this scheme, the optimality of the solution (in the sense of minimum time for this specific problem) is somewhat compromised, in exchange for smooth control and state histories, which are easier to implement in practice.

Both of the previous problems were solved using the DENMRA, starting from a uniform mesh of 25 points and ending with 50 points after 3 iterations. The results are shown in Figs. 9-15. In the first case, the airport is only 78 km away, so the aircraft must perform a large detour in order to reduce its altitude, yet too large a detour will unnecessarily increase the flight time. In the second case, since the distance to the airport

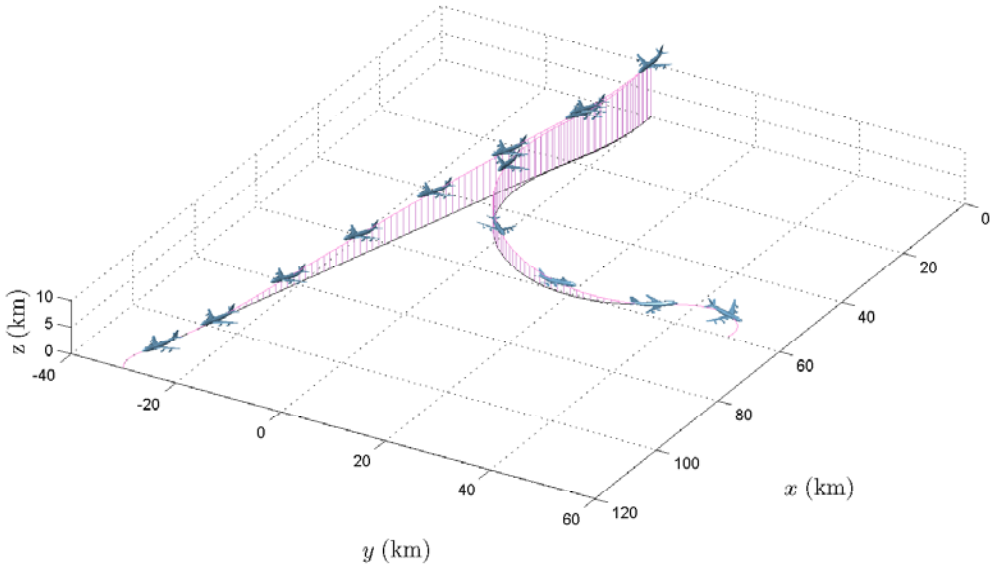


Figure 9. 3D minimum time landing trajectories.

is longer, the aircraft first turns slightly right, and pursues a stable descent at a small flight path angle with constant airspeed in the middle of the flight in order to fly over a long distance. The aircraft then performs a gentle left turn at the end of the flight to get aligned with the runway. The optimal trajectories for both cases are compared in Figs. 9 and 10, and the optimal state and control histories are shown in Figs. 11-15.

V.C. Minimum-Time Landing with Different Maximum Engine Power

In this section we investigate the problem of low thrust landing. Five minimum-time landing cases are computed using the DENMRA, with the upper bounds of throttle at 0%, 25%, 50%, 75% and 100% of the maximum engine thrust, respectively. The initial conditions are: $x(t_0) = 0\,\text{km}$, $y(t_0) = 0\,\text{km}$, $z(t_0) = 10\,\text{km}$, $v(t_0) = 240\,\text{m/s}$, $\gamma(t_0) = 0°$, $\psi(t_0) = 0°$, $\alpha(t_0)$ is free, $\phi(t_0) = 0°$ and $T(t_0)$ is free. The final conditions are: $x(t_f) = 110\,\text{km}$, $y(t_f) = -60\,\text{km}$, $z(t_f) = 0\,\text{km}$, $v(t_f) = 95\,\text{m/s}$, $\gamma(t_f) = -3°$, $\psi(t_f) = -45°$, $\alpha(t_f)$ is free, $\phi(t_f) = 0°$ and $T(t_f)$ is free. The horizontal distance between the initial position of the aircraft and the airport is 125.3 km. The length of the straight line connecting the initial position of the aircraft and the airport is 125.7 km. The results computed using the DENMRA are shown in Figs. 16-23.

As shown in Figs. 16-23, the aircraft initially increases the airspeed to travel along the path as fast as possible. If the thrust limit is greater than or equal to 25% of the maximum value, then the aircraft can reach the upper limit of the airspeed, which is 260 m/s. In the mean time, the thrust gradually decreases to zero and the path angle decreases. Then the aircraft almost takes a gliding descent, except that some thrust is used at the middle of the flight to keep the airspeed at its upper limit. At the final approach phase, the aircraft basically makes a powerless level flight and decreases its airspeed to the desired value before touch down. It is interesting to note that the thrust does not have too much impact on the minimum time value in this case. The time of the zero thrust landing is 603.8 s, and the time for 100% maximum throttle is 565.1 s. The aircraft never uses more than 27% of its maximum thrust except at the very beginning of the flight. Besides, the projections of all trajectories on the horizontal plane are almost identical. When thrust is allowed, the vertical profiles of the trajectories show a slightly deeper descent than the gliding case, which results in higher speed and hence a smaller landing time.

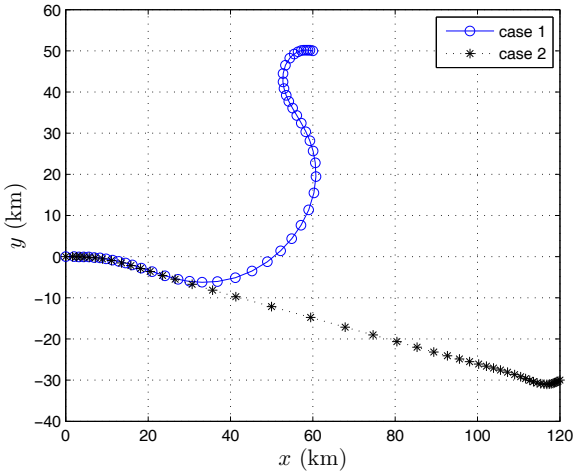
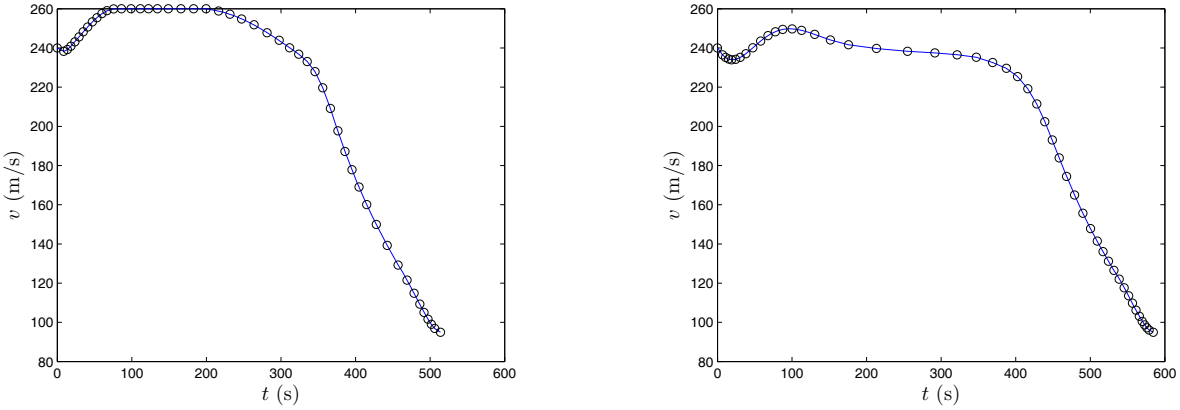
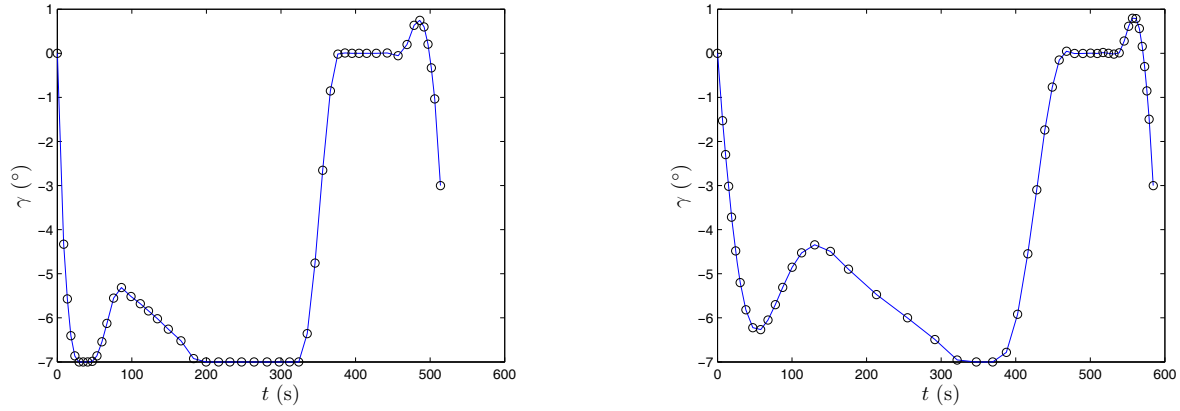


Figure 10. Trajectory profiles in the horizontal plane.



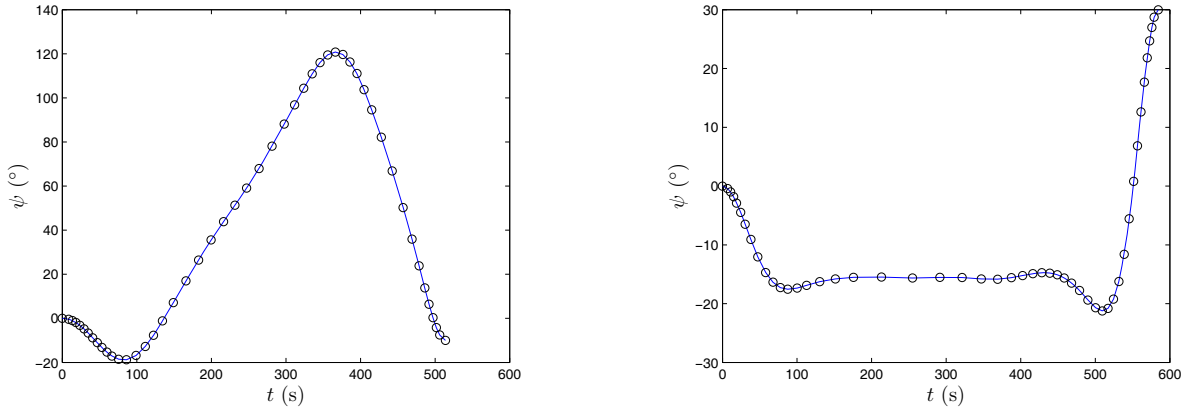
(a) Case 1 (b) Case 2

Figure 11. Airspeed histories for the two cases.



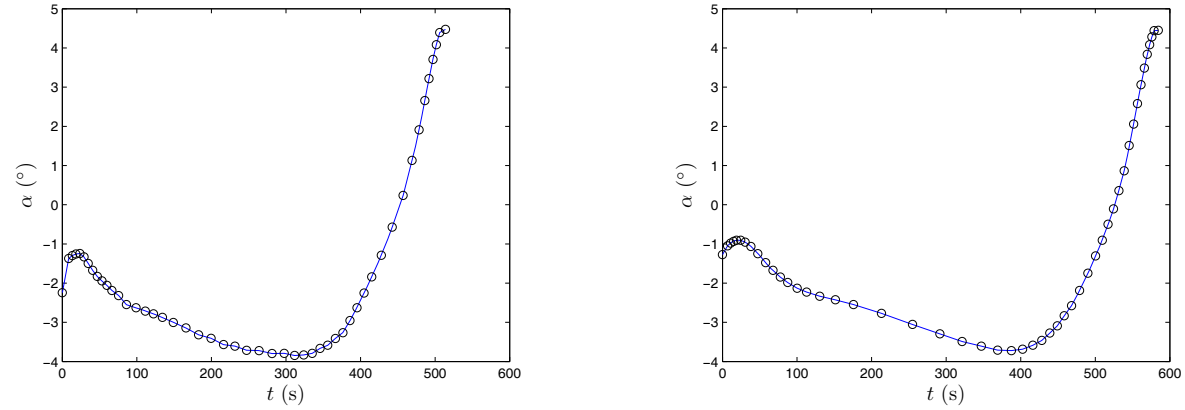
(a) Case 1 (b) Case 2

Figure 12. Path angle histories for the two cases.



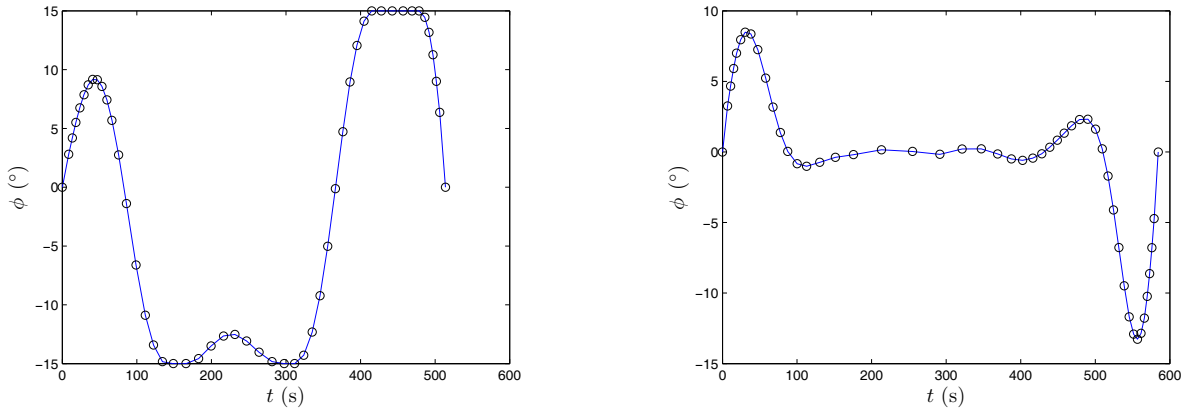
(a) Case 1 (b) Case 2

Figure 13. Heading angle histories for the two cases.



(a) Case 1 (b) Case 2

Figure 14. Angle of attack histories for the two cases.



(a) Case 1 (b) Case 2

Figure 15. Bank angle histories for the two cases.

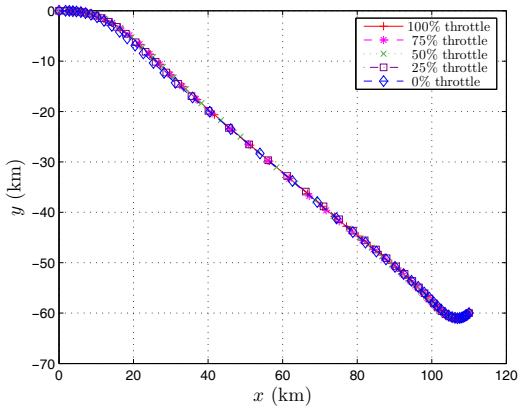


Figure 16. Horizontal trajectory profiles.

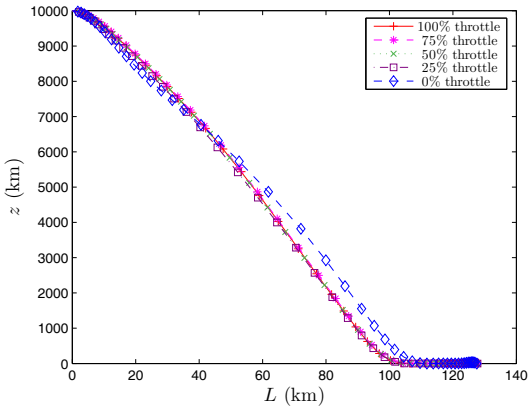


Figure 17. Vertical trajectory profiles.

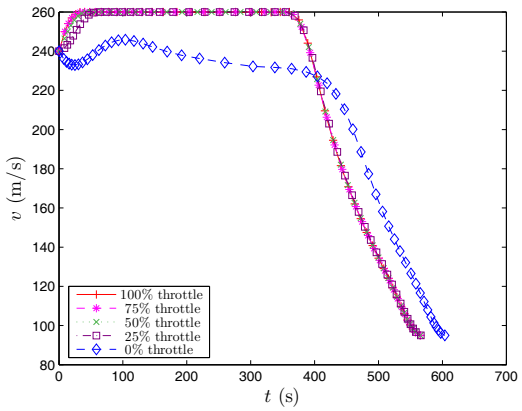


Figure 18. Airspeed.

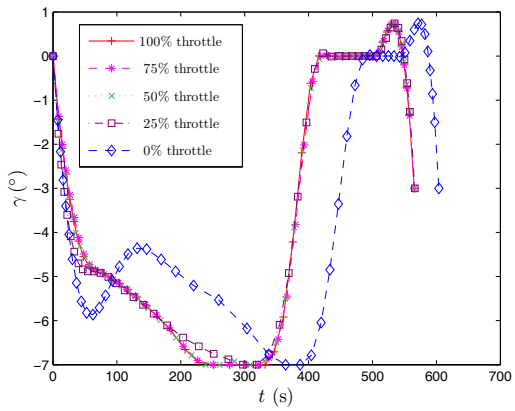


Figure 19. Flight path angle.

VI. Conclusions

It is known that multiresolution gridding methods can help convergence of numerical schemes while using wisely the available computational resources. Although multigrid/multiresolution methods have been very popular for solving problems in computational fluid dynamics, structural mechanics, etc, their use for solving optimal control problems is still very much at the early stages of development. In this paper we propose a general framework of mesh refinement for discretizing optimal control problems. The algorithm incorporates a density function which describes the distribution of the grid points in the mesh, and generates a non-uniform mesh with an appropriate distribution of points. As a specific instance of a suitable choice for a density function we propose one based on the local curvature of the solution. Simulation results show that the DENMRA automatically maintains an appropriate local level of discretization over the whole control and state time histories using the density function, which is computed from the results of the previous iteration. The grid generation is much easier than that of a previously proposed method (MTOA), while maintaining high numerical accuracy for the overall solution. The proposed algorithm is applied to solve several optimal trajectory landing problems for a large commercial aircraft.

Acknowledgment: This work has been supported by NASA with contract no. NNX08AB94A, monitored by NASA Ames Research Center. Contract monitor is Corey Ippolito.

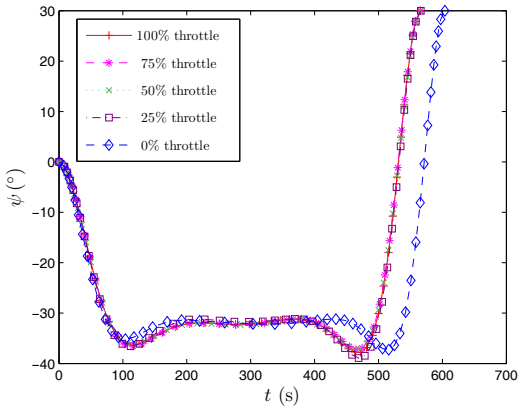


Figure 20. Heading angle.

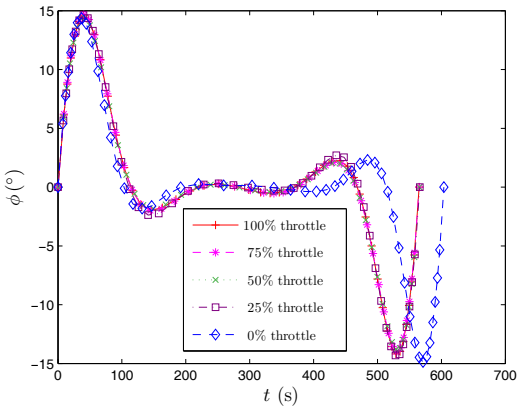


Figure 21. Bank angle.

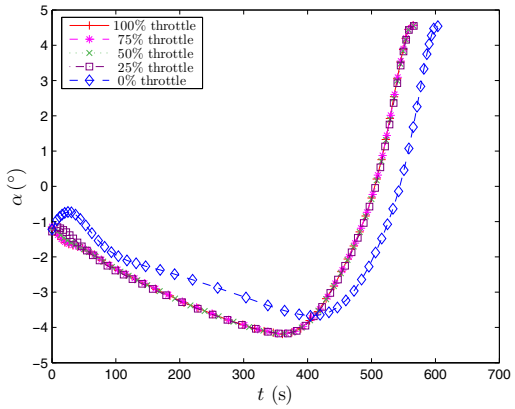


Figure 22. Angle of attack.

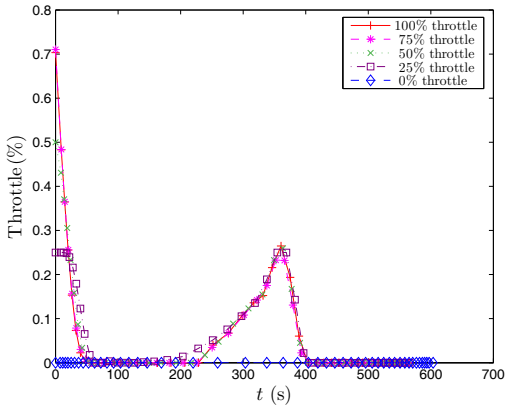


Figure 23. Thrust.

References

[1]Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193-207.

[2]Huntington, G., and Rao, A., "Comparison of Global and Local Orthogonal Collocation Methods for Solving Optimal Control Problems," *Proceedings of the 2007 American Control Conference*, July 2007, pp. 1950-1957.

[3]Chryssoverghi, I., Coletsos, J. and Kokkinis, B., "Discretization Methods for Optimal Control Problems with State Constraints," *Journal of Computational and Applied Mathematics*, Vol. 191, 2006, pp. 1-31.

[4]Jacques, Vlassenbroeck, "A Chebyshev Polynomial Method for Optimal Control with State Constraints," *Automatica*, Vol. 24, No. 4, 1988, pp. 499-506.

[5]Polak, E., *Optimization: Algorithms and Consistent Approximations,* Springer, New York, 1997.

[6]Gong, Q., Fahroo, F., and Ross, I.M., "Spectral Algorithm for Pseudospectral Methods in Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 460-471.

[7]Dontchev, A. L., Hager, W. W., and Malanowski, K., "Error Bounds for Euler Approximation of a State and Control Constrained Optimal Control Problem," *Numerical Functional Analysis and Optimization*, Vol. 21, 2000, pp. 653-682.

[8]Dontchev, A. L., Hager, W. W., Veliov, V. M., "Second-Order Runge-Kutta Approximations in Control Constrained Optimal Control," *SIAM Journal on Numerical Analysis*, Vol. 38, No. 1, 2001, pp. 202-226.

[9]Chryssoverghi, I., Coletsos, J. and Kokkinis, B., "Approximate Gradient Projection Method with Runge-Kutta Schemes for Optimal Control Problems," *Computational Optimization and Applications*, Vol. 29, 2004, pp. 91-115.

[10]Schwartz, A.L., *Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems*, Ph.D. Thesis, Berkeley, University of California, 1996.

[11]Gong, Q., Kang, W., and Ross, I. M., "A Pseudospectral Method for the Optimal Control of Constrained Feedback Linearizable Systems," *IEEE Transactions on Automatic Control*, Vol. 51, No. 7, July 2006, pp. 1115-1129.

[12]Jain, S. and Tsiotras, P., "Trajectory Optimization Using Multiresolution Techniques," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, pp.1424-1436.

[13]Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-scale Constrained Optimization," Numerical Analysis Report 97-2, Department of Mathematics, University of California, San Diego, La Jolla, CA, 1997.

[14]Bryson, A. E., Ho, Y., *Applied Optimal Control-Optimization, Estimation and Control,* Hemispere Publishing Corporation, Washington, 1975.

[15]Folland, G. B., *Real Analysis: Modern Techniques and Their Applications,* John Wiley & Sons, Inc, Hoboken, 1999.

[16]Earth atmosphere model. (2008). Retrieved Feb 15, 2009, from http://www.grc.nasa.gov/WWW/K-12/airplane/atmosmet.html.

[17]Zhao.,Y.J., "Optimal patterns of glider dynamic soaring," *Optimal Control Applications and Methods*, Vol. 24, No. 2, pp. 67-89., 2004.

[18]Parker, W. V. and Pryor, J. E., "Polygons of Greatest Area Inscribed in an Ellipse," *The American Mathematical Monthly*, Vol. 51, No. 4, 1944, pp. 205-209.

Appendix: Piecewise Linear Approximation of C^2 a.e. Functions in the $\mathcal{L}^1$-norm

The fact that a function $\Gamma : \mathcal{I} \to \mathbb{R}$ is C^2 a.e., where $\mathcal{I} = [t_a, t_b] \subset \mathbb{R}$, implies that its intrinsic curvature is continuous a.e.. Recall that the curvature κ of Γ is given by:

$$\kappa(t) = \frac{|\Gamma''(t)|}{|(1+\Gamma'^2(t))^{3/2}|},$$

where $\Gamma'' = \frac{\mathrm{d}^2\Gamma}{\mathrm{d}t^2}$ and $\Gamma' = \frac{\mathrm{d}\Gamma}{\mathrm{d}t}$. Let Γ be bounded, with curvature κ defined a.e.. The natural coordinate s defined by $s(t) = \int_{t_a}^{t} \left[1+\Gamma'^2(t)\right]^{\frac{1}{2}} \mathrm{d}t$ is a measure of the length of the curve defined by the graph of Γ.

Let $T_{\mathcal{I},N} = \{t_i\}_{1\le i\le N}$ be a partition of the interval $\mathcal{I}$ using N points, where $t_a = t_1 < t_2 < \ldots < t_N = t_b$.

Definition A.1. The function $P_{T_{\mathcal{I},N}} : \mathcal{I} \to \mathbb{R}$ is a piecewise linear approximation of Γ on the interval $\mathcal{I}$ over the partition $T_{\mathcal{I},N}$ if, for any $t \in [t_i, t_{i+1})$ with $1 \le i \le N-1$ and $t_i, t_{i+1} \in T_{\mathcal{I},N}$, the following equation holds:

$$P_{T_{\mathcal{I},N}}(t) = \Gamma(t_i) + \frac{t-t_i}{t_{i+1}-t_i}\left(\Gamma(t_{i+1}) - \Gamma(t_i)\right),$$

where $P_{T_{\mathcal{I},N}}(t_N) = \Gamma(t_N)$.

Definition A.2. The function $P_{T''_{\mathcal{I},N}} : \mathcal{I} \to \mathbb{R}$ is a refinement of $P_{T'_{\mathcal{I},N}}$ if $\|\Gamma - P_{T''_{\mathcal{I},N}}\|_1 < \|\Gamma - P_{T'_{\mathcal{I},N}}\|_1$, where $\|\cdot\|_1$ denotes the $\mathcal{L}^1$-norm defined by $\|f\|_1 = \int_{t_a}^{t_b} |f(t)|\,\mathrm{d}t$.

Definition A.3. $P_{T^*_{\mathcal{I},N}}$ is the best piecewise linear approximation of Γ with N points if $\|\Gamma - P_{T^*_{\mathcal{I},N}}\|_1 < \|\Gamma - P_{T_{\mathcal{I},N}}\|_1$ for any $T_{\mathcal{I},N} \neq T^*_{\mathcal{I},N}$.

Definition A.4. A density function on Γ is a bounded mapping $\rho_\Gamma : \mathcal{I} \to \mathbb{R}_+$ which is Lebesgue integrable on $\mathcal{I}$.

With the density function ρ_Γ defined a.e. on $\mathcal{I}$, for any $N \ge 2$, the grid points denoted by $\{(t_i, \Gamma(t_i))\}_{i=1}^N$ are allocated on Γ such that $t_1 = t_a$, and

$$\int_{t_a}^{t_i} \rho_\Gamma(\tau)\,\mathrm{d}\tau = \frac{i-1}{N-1}. \tag{A.1}$$

Remark A.1. The density function ρ_Γ describes the distribution pattern of the grid points along Γ, by which a piecewise linear approximation of Γ can be determined.

Definition A.5. The mass function on Γ is a continuously differentiable mapping $m_\Gamma : \mathcal{I} \to \mathbb{R}_+$ defined by

$$m_\Gamma(t) = \int_{t_a}^{t} \rho_\Gamma(\tau)\,\mathrm{d}\tau, \tag{A.2}$$

where $t \in \mathcal{I}$.

Remark A.2. The mass function m_Γ also describes the distribution of grid points, with $m_\Gamma(t_i) = (i-1)/(N-1)$. This function is introduced to facilitate the proof of Theorem A.2.

Proposition A.1. The best piecewise linear approximation of a function Γ with nonzero constant curvature κ using three points is given by $P_{T^*_{\mathcal{I},3}}$, where $T^*_{\mathcal{I},3} = \{t_1, t_2, t_3\}$ with $t_1 = t_a$, $t_3 = t_b$ and

$$\|(t_1, \Gamma(t_1)) - (t_2, \Gamma(t_2))\| = \|(t_3, \Gamma(t_3)) - (t_2, \Gamma(t_2))\|,$$

where $\|\cdot\|$ is the Euclidean norm. In other words, the points are evenly distributed along the arc Γ.

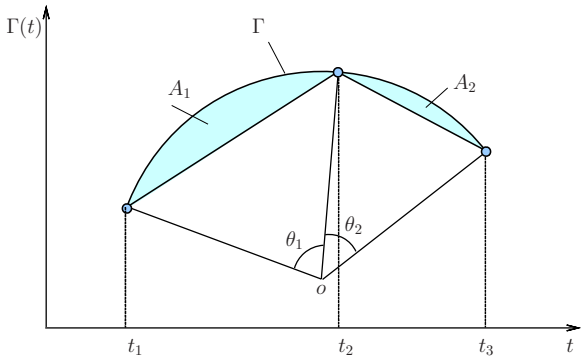


Figure 24. Approximation error in terms of the $\mathcal{L}^1$-norm, for a curve Γ of constant curvature.

Proof. The graph of a function with constant nonzero curvature is a circular arc, as shown in Fig. 24, with o denoting the center of the corresponding circle. The shaded area A_1 is given by:

$$A_1 = \frac{\theta_1}{2}\kappa^{-2} - \kappa^{-1}\sin\left(\frac{\theta_1}{2}\right)\kappa^{-1}\cos\left(\frac{\theta_1}{2}\right) = \frac{\theta_1}{2}\kappa^{-2} - \frac{1}{2}\kappa^{-2}\sin\theta_1 = \frac{1}{2}\kappa^{-2}\left(\theta_1 - \sin\theta_1\right).$$

Similarly, the shaded area A_2 is given by:

$$A_2 = \frac{1}{2}\kappa^{-2}\left(\theta_2 - \sin\theta_2\right).$$

The approximation error ξ of the piecewise linear approximation of Γ using the $\mathcal{L}^1$-norm is given by:

$$\xi = A_1 + A_2 = \frac{1}{2}\kappa^{-2}(\theta_1 + \theta_2 - \sin\theta_1 - \sin\theta_2) = \frac{1}{2}\kappa^{-2}\big(\theta - \sin\theta_1 - \sin(\theta - \theta_1)\big),$$

where $\theta = \theta_1 + \theta_2 = s(t_3)\kappa$, which is constant for the given Γ. The first order derivative of ξ with respect to θ_1 is given by:

$$\frac{\mathrm{d}\xi}{\mathrm{d}\theta_1} = -\frac{1}{2}\kappa^{-2}\big(\cos\theta_1 - \cos(\theta - \theta_1)\big).$$

Since in any discretization of an optimal control problem, N is always large enough, the inequalities $\theta_1 < \pi$ and $\theta_2 < \pi$ hold. The first order necessary condition for the minimization of ξ, $\frac{\mathrm{d}\xi}{\mathrm{d}\theta_1} = 0$, yields that $\theta_1 = \theta_2$. This is indeed a minimum since

$$\frac{\mathrm{d}^2\xi}{\mathrm{d}\theta_1^2} = \frac{1}{2}\kappa^{-2}(\sin\theta_1 + \sin\theta_2) > 0.$$

Hence the proposition is proved. □

Theorem A.1. The best piecewise linear approximation of a function Γ with constant nonzero curvature κ on a bounded interval $\mathcal{I}$ yields a constant density along the curve.

Proof. Since there is a one-one correspondence between the points of $P_{T^*_{\mathcal{I},N}}$ (except the first one) and the angles $\theta_1, \theta_2, \ldots, \theta_{N-1}$ (see Fig. 24), the result is equivalent to the assertion that for any function Γ with nonzero constant κ, the best piecewise linear approximation $P_{T^*_{\mathcal{I},N}}(t)$ corresponds to $\theta^* = (\theta_1^*, \theta_2^*, \ldots, \theta_{N-1}^*)$,

where $\theta_1^* = \theta_2^* = \ldots = \theta_{N-1}^* = s(t_b)\kappa/(N-1)$ for any $N \geq 2$. Let $\theta = (\theta_1, \theta_2, \ldots, \theta_{N-1}) \in [0,\pi]^{N-1}$. Define mappings $g_i : [0,\pi]^{N-1} \to [0,\pi]^{N-1}$ such that

$$g_i(\theta) = g_i(\theta_1, \theta_2, \ldots, \theta_i, \theta_{i+1}, \ldots, \theta_{N-1}) = (\theta_1, \theta_2, \ldots, \frac{\theta_i + \theta_{i+1}}{2}, \frac{\theta_i + \theta_{i+1}}{2}, \ldots, \theta_{N-1}),$$

for any $1 \leq i < N-1$, and construct a mapping $g : [0,\pi]^{N-1} \to [0,\pi]^{N-1}$ by $g = g_{N-1} \circ g_{N-2} \circ \ldots \circ g_1$. For any $\theta \neq \theta^*$, $g(\theta)$ is a refinement of θ following Proposition A.1, so θ^* corresponds to $P_{T^*_{\mathcal{I},N}}(t)$ by definition. For a function Γ with constant curvature, θ^* represents a uniform distribution of points along Γ. As this is true for arbitrary N, $\rho_\Gamma(s)$ must be constant along Γ, hence the theorem is proved. □

Theorem A.2. Suppose a function Γ consists of two parts Γ_1 and Γ_2, with constant curvature κ_1 and κ_2 defined on each part, respectively. Then the best piecewise linear approximation of Γ is achieved with constant density ρ_{Γ_1} and ρ_{Γ_2} on Γ_1 and Γ_2, respectively, satisfying

$$\frac{\rho_{\Gamma_1}}{\rho_{\Gamma_2}} = \left(\frac{\kappa_1}{\kappa_2}\right)^{\frac{1}{3}}.$$

More generally, for a function Γ with arbitrary curvature κ bounded a.e., the best piecewise linear approximation of Γ corresponds to a density function ρ_Γ with $\rho_\Gamma(s) = c\kappa(s)^{1/3}$, where c is a constant.

Proof. Since the points allocated to Γ_1 and Γ_2 must constitute the best piecewise linear approximation of Γ_1 and Γ_2, respectively, we first consider the approximations of Γ_1 and Γ_2 separately. Let $\theta_1 = \kappa_1 S_1$, where S_1 is the length of Γ_1. With the best piecewise linear approximation of function Γ_1 using N_1 points, the total approximation error ξ_{N_1} in the $\mathcal{L}^1$-norm is given by:

$$\xi_{N_1} = \kappa_1^{-2} \sum_1^{N_1-1} \left(\frac{\theta_1}{(N_1-1)} - \sin\frac{\theta_1}{N_1-1}\right) = \kappa_1^{-2}\left(\theta_1 - (N_1-1)\sin\frac{\theta_1}{N_1-1}\right). \tag{A.3}$$

Taking the Taylor series expansion, one obtains

$$\sin\frac{\theta_1}{(N_1-1)} \approx \frac{\theta_1}{(N_1-1)} - \frac{1}{6}\left(\frac{\theta_1}{N_1-1}\right)^3 + \frac{1}{120}\left(\frac{\theta_1}{N_1-1}\right)^5.$$

Bringing the first two terms into Eq. (A.3), and substituting $(N_1-1)/\rho_{\Gamma_1}$ with S_1, we have

$$\begin{aligned}
\xi_{N_1} &= \frac{1}{6}\kappa_1^{-2}\frac{\theta_1^3}{(N_1-1)^2} = \frac{1}{6}\kappa_1^{-2}(\kappa_1 S_1)^3\frac{1}{(N_1-1)^2} \\
&= \frac{1}{6}\frac{\kappa_1 S_1^3}{(N_1-1)^2} = \frac{1}{6}\kappa_1 S_1 \rho_{\Gamma_1}^{-2} \\
&= \frac{1}{6}\kappa_1 S_1^3 m_{\Gamma_1}^{-2},
\end{aligned}$$

where m_{Γ_1} is the integral of ρ_{Γ_1} along Γ_1. Since ρ_{Γ_1} is constant along Γ_1 by Theorem.A.1, $m_{\Gamma_1} = \rho_{\Gamma_1} S_1 = N_1 - 1$.

Similarly, the best piecewise linear approximation of Γ_2 using N_2 points yields

$$\xi_{N_2} = \frac{1}{6}\kappa_2 S_2^3 m_{\Gamma_2}^{-2}.$$

The total approximation error is thus given by:

$$\xi = \xi_1 + \xi_2 = \frac{1}{6}\left(\kappa_1 S_1^3 m_{\Gamma_1}^{-2} + \kappa_2 S_2^3 m_{\Gamma_2}^{-2}\right).$$

When the total number of grid points is fixed, $m_\Gamma = m_{\Gamma_1} + m_{\Gamma_2}$ is a constant. Moving a point from Γ_1 to Γ_2 or vice versa is equivalent to a change of m_{Γ_1} while keeping m_Γ constant. The definition of the best

piecewise linear approximation of Γ requires that $\frac{\mathrm{d}\xi}{\mathrm{d}m_{\Gamma_1}} = 0$, hence we have

$$\begin{aligned} 0 &= \frac{\mathrm{d}\xi}{\mathrm{d}m_{\Gamma_1}} = \frac{\mathrm{d}\xi_1}{\mathrm{d}m_{\Gamma_1}} + \frac{\mathrm{d}\xi_2}{\mathrm{d}m_{\Gamma_1}} = \frac{\mathrm{d}\xi_1}{\mathrm{d}m_{\Gamma_1}} + \frac{\mathrm{d}\xi_2}{\mathrm{d}m_{\Gamma_2}}\frac{\mathrm{d}m_{\Gamma_2}}{\mathrm{d}m_{\Gamma_1}} \\ &= \frac{\mathrm{d}\xi_1}{\mathrm{d}m_{\Gamma_1}} - \frac{\mathrm{d}\xi_2}{\mathrm{d}m_{\Gamma_2}} = -\frac{1}{3}\kappa_1 S_1^3 m_{\Gamma_1}^{-3} + \frac{1}{3}\kappa_2 S_2^3 m_{\Gamma_2}^{-3} \\ &= -\frac{1}{3}\kappa_1 \rho_{\Gamma_1}^{-3} + \frac{1}{3}\kappa_2 \rho_{\Gamma_2}^{-3}, \end{aligned}$$

which implies

$$\frac{\rho_{\Gamma_1}}{\rho_{\Gamma_2}} = \left(\frac{\kappa_1}{\kappa_2}\right)^{\frac{1}{3}}.$$

For the more general case when the curvature κ of Γ is bounded a.e., as shown in Fig. 25, by taking arbitrarily small pieces of Γ, κ can be treated as constant on each piece. Using the former result, the best piecewise linear approximation of Γ yields $\rho_\Gamma(s) = c\kappa(s)^{1/3}$, where c is a constant such that Eq. (A.1) is satisfied. □

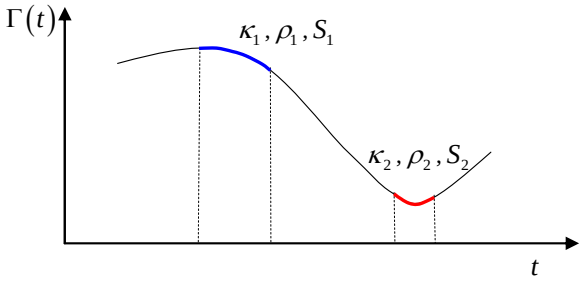


Figure 25. Density function for the best piecewise linear approximation of an arbitrary C^2 function Γ.

Remark A.3. It can be shown that the theory of polygons of the greatest area inscribed in an ellipse[18] is a special case of Theorem A.2, with Γ being part of an ellipse.

Remark A.4. When the control history is discontinuous, it can still be approximated by a C^2 a.e. function Γ with a large curvature around the points of discontinuity. Hence, the value of the corresponding density function is also large around these discontinuities, and more points will be distributed around discontinuities during the mesh refinement process. Furthermore, because ρ_Γ depends on the curvature of Γ, which is calculated using the first and second derivatives of Γ, ρ_Γ is also sensitive to the first and second order discontinuities in the control history. This helps capture the higher order smoothness irregularities such as corners.