JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 21, No. 5, September–October 1998

Collocation Versus Differential Inclusion in Direct Optimization

B. A. Conway* and K. M. Larson†
University of Illinois at Urbana–Champaign, Urbana, Illinois 61801

In the so-called direct method of solution of optimal control problems, either the state variable time history or the control variable time history, or both, of the continuous problem are discretized. The problem then becomes a parameter optimization problem. The system-governing equations may be satisfied by explicit numerical integration or implicitly, by including nonlinear constraints, which are in fact quadrature rules. A method termed differential inclusion has been recommended for the solution of certain classes of such problems because it reduces the size of the parameter optimization problem. It does this by removing bounded control variables in favor of bounds on attainable time rates of change of the states. The smaller problem is then in principle solved more quickly and reliably. We demonstrate analytically and with several computed problem solutions that differential inclusion, because it requires the use of an implicit quadrature rule with the lowest possible order of accuracy, i.e., Euler's rule, yields larger rather than smaller nonlinear programming problems than direct methods, which retain the control variables but use much more sophisticated implicit quadrature rules.

Introduction

DIRECT methods for the solution of continuous optimal control problems are methods that do not explicitly employ the necessary conditions for optimality. They are becoming widely used for many reasons, the principal one being that the indirect approach, which introduces Lagrange multipliers (or costates) and yields a two-point-boundary-value problem, is very difficult to solve for all but the simplest problems. In the direct method the control time history and/or the state variable time history is discretized. Satisfaction of the system-governing equations can be ensured in several ways.

In the POST software[1] only the control time history is discretized. The system-governing equations are integrated forward numerically using the discrete control time history. The problem becomes a nonlinear programming (NLP) problem if, as is typically the case, the initial and/or terminal constraints or the performance index are nonlinear functions of the states and controls. Collocation-based methods discretize both the control and state variable time histories, i.e., the states and controls are known only at discrete points, typically the beginning and end of each time segment into which the total time is subdivided (referred to as the nodes of the discretization). The system-governing equations are satisfied by including nonlinear constraint equations (the defects) for each state and in each time segment with the constraints representing quadrature across the time segment. The collocation methods thus integrate the system equations implicitly. Compared to the method used in POST, collocation methods typically use many more NLP variables and constraint equations but because many thousands of trajectories do not have to be integrated numerically significant execution time is saved.

The OTIS program[2] uses collocation and assumes the state time histories can be approximated by cubic polynomials in time within each time segment. Enforcing satisfaction of the system equations at the left and right nodes of the segment and at the center of the segment yields a constraint that is equivalent to quadrature across the segment using Simpson's rule. Higher-order Gauss–Lobatto quadrature rules with greater order of accuracy have been developed by Herman and Conway[3] for use with collocation and have proved successful with a variety of problems. Runge–Kutta type quadrature has also been successfully used with collocation by Enright and Conway[4] and by Scheel and Conway.[5] The attraction of implicit integration rules with higher order of accuracy for use with collocation is precisely the same as it is for numerical integration; when using quadrature rules with smaller error, the step size can be increased without sacrificing accuracy. When larger step sizes, i.e., fewer segments, are used in the discretization, the number of variables in the NLP problem decreases. Because the execution time and the difficulty of solving the NLP problem increase geometrically with increasing problem size there is an obvious advantage to using accurate, high order of accuracy implicit quadrature rules with collocation.

Another direct method that uses implicit integration of the system equations is differential inclusion. In differential inclusion only the state variable time history is discretized; bounded controls are eliminated and replaced with bounds on attainable values of the state variable time rates of change. An elementary implicit integration rule is then used to write the time rates of change as functions of the state variables alone. Because the control variables are eliminated the number of variables in the resulting NLP problem is smaller for differential inclusion than for collocation, for a given discretization of the time history.

In a recent paper, "Should Controls Be Eliminated While Solving Optimal Control Problems via Direct Methods?," Kumar and Seywald[6] advocate doing precisely what their singularly specific title suggests, for at least the class of optimal control problems with linearly appearing bounded controls. This recommendation is based on an analysis quantifying the saving in number of NLP parameters as earlier mentioned and on the successful solution of two test problems on which both methods are used.

We consider the question again. How collocation and differential inclusion are applied to an optimal control problem will be briefly described. Then the number of NLP parameters resulting from the application of each method to a given problem will be compared, but with a different criterion: How many NLP parameters are required with each method to achieve the same accuracy of the discrete approximation to the continuous optimal trajectory? Three example optimal control problems will be solved with both methods; one (F-15 minimum time-to-climb) is essentially the same as that used in Ref. 6, and the other two are chosen because their optimal solution can be obtained analytically so that the accuracy of each discrete approximation can be determined exactly.

Collocation Approach

There are several different ways to discretize the problem time history and several ways to ensure satisfaction of the system governing equations. It is not possible to describe all of the available methods here; the interested reader can find several collocation methods

Received Nov. 13, 1997; revision received Feb. 23, 1998; accepted for publication April 20, 1998.

*Professor, Department of Aeronautical and Astronautical Engineering. E-mail: bconway@uiuc.edu. Associate Fellow AIAA.

†Graduate Research Assistant, Ph.D. Candidate, Department of Aeronautical and Astronautical Engineering.

described in Refs. 2–4 and 7–9. However, to clearly differentiate collocation from differential inclusion, one popular discretization, that used in the OTIS program, will be described here.

In this method of collocation, introduced by Hargraves and Paris,[2] the time history is divided into segments, the ith segment having width Δt_i. The system state and control variables are assumed known only at the left and right sides (the nodes) of each segment; the system governing equations may then be evaluated at each node, i.e., let $f_i = \dot{x}(t_i)$. The four quantities, i.e., the states and their time rates of change at the two boundaries of the segment $(x_i, x_{i+1}, f_i, f_{i+1})$, determine uniquely a cubic polynomial approximation for the state within the segment. By forcing the slope of the cubic to equal the system time rate of change evaluated at the center of the segment, i.e., by requiring $f_c = \dot{x}((t_i + t_{i+1})/2, u_c)$, a constraint is created on successive x:

$$\int_{t_i}^{t_{i+1}} f(t)\,\mathrm{d}t \approx \frac{\Delta t_i}{6}[f(t_i) + 4f(t_c) + f(t_{i+1})] \tag{1}$$

where

$$t_c = \frac{t_i + t_{i+1}}{2} \tag{2}$$

This, of course, is just Simpson's quadrature rule.[9] If the total time history is divided into N such segments and if there are n system state variables and m system control variables, then the discretized system will be approximated using $(N+1) \times (n+m)$ parameters. (If the center control u_c is free, i.e., not determined by the left and right values of the control, then there will be $N \times m$ additional parameters.) If the vectors containing all of the state and all of the control parameters are represented by $\boldsymbol{x}$ and $\boldsymbol{u}$, respectively, then the continuous problem is converted into a nonlinear programming problem: Minimize $\phi(t_f, \boldsymbol{x}, \boldsymbol{u})$ subject to $n \times N$ nonlinear constraint equations of form $C_i(x_i, u_i, \Delta t_i) = 0$.

Of course, for a particular problem there are likely to be additional nonlinear and linear constraints involving the states and controls. If all of the constraint equations (1) are satisfied by the NLP problem solver, then the system governing equations have been implicitly integrated using Simpson's rule, which has a local truncation error proportional to $(\Delta t_i)^5$ and, thus, an order of accuracy of four. The Simpson's rule constraints (1) are just one choice among many. Enright and Conway[4] introduced the use of parallel shooting with collocation using a constraint that is equivalent to quadrature using a fourth-order Runge–Kutta process. This form of collocation is particularly suited for large problems, such as the optimal long-duration (100 revolution) low-Earth-orbit–geosynchronous-Earth-orbit (LEO-GEO) raising problem solved by Scheel and Conway.[5] Higher-order Gauss–Lobatto quadrature rules were introduced as collocation constraints by Herman and Conway.[3] These rules have very high order of accuracy, e.g., the fifth-degree Gauss–Lobatto rule constraint has a local truncation error proportional to $(\Delta t_i)^9$ and, thus, an order of accuracy of 8.

Differential Inclusion Approach

The basic concept underlying the method of differential inclusion is that bounds on the control variables of the problem yield bounds on the attainable rates of change of the states. If the problem is such as to allow the control variables to be written explicitly as functions of the states and their time rates of change, which is straightforwardly accomplished only when the problem is linear in the controls, then the control variables can be completely eliminated from the problem. The resulting differential inclusion problem is qualitatively similar to the collocation problem; the control variables are absent, but both problems require some form of quadrature rules to relate adjacent states, and both problems may include additional linear and nonlinear constraints on the state variables. Elimination of the discrete controls ostensibly reduces the dimensionality of the differential inclusion NLP problem, compared to using collocation.[6,10]

The method is best described by example. Collocation and differential inclusion solutions to three simple problems will be compared. One of these problems is that of steering an F-15 aircraft using bounded throttle setting $\eta(t)$ and bounded vertical load factor $n(t)$ from given initial conditions to the farmost point to the right of the level flight envelope (dash-point) in minimum time. The problem is chosen in part because it is the example used by Kumar and Seywald in their comparison[6] of differential inclusion and collocation. The problem may be stated as: Minimize final time t_f subject to

$$\dot{h} = v \sin\gamma \tag{3}$$

$$\dot{E} = (\eta T - D)(v/mg) \tag{4}$$

$$\dot{\gamma} = (g/v)(n - \cos\gamma) \tag{5}$$

where h is altitude, E is specific energy, γ is flight-path angle, and the velocity is given by $v = \sqrt{[2g(E-h)]}$. The control constraints are

$$0 \le \eta \le 1 \tag{6}$$

$$-n_{\max} \le n \le +n_{\max} \tag{7}$$

and the boundary conditions are

$$\begin{aligned} h(0) &= 5\,\mathrm{m}, & h(t_f) &= 12119.3\,\mathrm{m} \\ E(0) &= 2668\,\mathrm{m}, & E(t_f) &= 38029.2\,\mathrm{m} \\ \gamma(0) &= 0\,\mathrm{rad}, & \gamma(t_f) &= 0\,\mathrm{rad} \end{aligned} \tag{8}$$

The initial states represent the airplane shortly after takeoff, and the final states represent level flight conditions at the dash-point. The mass of the F-15 is assumed constant with the value $m = 16818$ kg, and the gravitational acceleration is assumed to be $g = 9.80665\,\mathrm{m/s^2}$. The atmospheric and drag models used may be found in Refs. 11–13. The thrust model of Refs. 11–13 was problematic and so propulsion performance as a function of altitude and Mach number were obtained by us directly from the manufacturer of the F-15.

This problem is linear in the control variables and so it is easy to solve for η and n and, hence, cast the problem in differential inclusion format, i.e.,

$$\dot{h}_i - \bar{v}_i \sin\bar{\gamma}_i = 0 \tag{9}$$

$$0 \le \frac{\dot{E}_i(mg/\bar{v}_i) + \bar{D}_i}{\bar{T}_i} \le 1 \tag{10}$$

$$-n_{\max} \le [\dot{\gamma}_i(\bar{v}_i/g) + \cos\bar{\gamma}_i] \le +n_{\max} \tag{11}$$

Constraints (9–11) require some approximation for the time rates of change of the states. In the differential inclusion approach, the mean value of a state (in the ith time segment) and the corresponding state rate are approximated by

$$\bar{x}_i = \frac{x_{i+1} + x_i}{2} \tag{12}$$

and

$$\dot{\bar{x}}_i = \frac{x_{i+1} - x_i}{\Delta t_i} \tag{13}$$

respectively. For a system employing N time segments, the NLP problem thus becomes, minimizing

$$t_f = \sum_{i=1}^{N} \Delta t_i$$

subject to $3N$ constraints, i.e., Eqs. (9–11) evaluated in each of N segments, using Eqs. (12) and (13) and the initial and terminal boundary conditions (8).

Advantages and Disadvantages of Differential Inclusion

The principal advantage claimed for the differential inclusion method is reduced problem size.[6,10] By eliminating the controls the discretized problem has $n \times (N+1)$ variables, as opposed to collocation (using Simpson's rule constraints), which will have $n \times (N+1) + m \times (N+1)$ variables. Using higher order of accuracy implicit quadrature rules appears to improve the advantage of differential inclusion with respect to problem size, e.g., using collocation with a fifth-degree Gauss–Lobatto quadrature rule yields $n \times (2N+1) + m \times (4N+1)$ variables. Because the CPU time required to solve NLP problems increases geometrically with problem size, this yields, in principle, a considerable speed advantage for differential inclusion. In addition, smaller problems are simply easier to solve so that the differential inclusion solution gains robustness because of its size.

The acknowledged disadvantages of differential inclusion are 1) a near restriction of the method to problems with linearly appearing controls, 2) much increased difficulty in the analytical derivation of the gradients of the constraints with respect to the problem parameters, which are needed by the NLP solver, and 3) the requirement of obtaining the control time histories, if needed a posteriori.

Do the advantages of the differential inclusion method outweigh the disadvantages? Kumar and Seywald do not unequivocally answer this question, which is just that posed in their paper's title.[6] They conclude that the user "... should choose to keep or eliminate the controls based on the problem at hand."[6] This is both true and sensible, though it does not give much guidance, especially for a potential user who might not be very familiar with direct solution methods. In the remainder of this paper we will show that the circumstances in which the control-elimination approach should be preferred are very narrow.

Quadrature Rules

In the preceding section it was shown how, for a given number of states n and controls m, the number of NLP variables resulting from applying the differential inclusion solution method is smaller than if collocation is used for a given discretization of the continuous time history into N segments. Completely ignored to this point is how N should be chosen. In Ref. 6 the problem size is compared assuming that N will be the same whether one uses collocation or differential inclusion,[6] but there is no reason this should be so.

N should be chosen to satisfy two requirements: N should be large enough, i.e., the time steps should be small enough, to yield sufficient accuracy of the implicit integration of the system governing equations and to adequately capture the character of the solution time history. For example, if the true solution time history for a state or control variable is a sinusoidal variation, each period of oscillation of this variable will need to be represented by several time segments. But once the latter requirement is fulfilled the accuracy of the implicit integration of the system governing equations is determined by the choice of the quadrature rule that is used by the method.

In differential inclusion, as exemplified by the work of Seywald[10] and of Kumar and Seywald,[6] the state rate is determined by Eq. (13), but of course this relationship may be written as

$$x_{i+1} = x_i + \dot{x}_i \Delta t_i \tag{14}$$

which is Euler's integration rule. The Euler step has an error term proportional to $(\Delta t_i)^2$ and, hence, has an order of accuracy of 1. It is easy to see that differential inclusion is limited to using explicit quadrature rules such as Euler's rule. If we try to use a more accurate rule, such as the trapezoid rule,

$$x_{i+1} = x_i + (\Delta t_i/2)[\dot{x}_i + \dot{x}_{i+1}] \tag{15}$$

which has order of accuracy of 2, there is no way to isolate the state rate at the ith node as a function only of discrete states. Thus, it is not possible to write the differential inclusion method constraints, Eqs. (9–11) being a good example, exclusively as a function of the system variables, i.e., the discrete states.

The collocation method is not limited in this way. As mentioned earlier, the collocation solution has been constructed using constraints based on many types of quadrature rules including Simpson's rule,[2,9] higher-order Gauss–Lobatto rules,[3] and even a four-stage Runge–Kutta (R–K) process.[4] These methods have incomparably better order of accuracy than Euler's rule, which is the least accurate quadrature rule there is. In addition, there is often no penalty paid in using these accurate rules. For example, when Kumar and Seywald[6] solve the F-15 minimum time-to-climb problem using collocation (for comparison to the same problem solved using differential inclusion), they use constraints based on the same Euler rule. But using the much more accurate Simpson's rule (1) would not have increased the dimensionality of the problem at all. Using collocation with the four-stage R–K process can actually decrease the dimensionality of the problem significantly because, although intermediate values of the states are estimated by the R–K process, they are not NLP variables; thus, the solution receives the benefits of a finer discretization without the increase in problem size that usually accompanies it.[4]

The problem faced by the user is, thus, entirely analogous to the problem faced in choosing a method for numerical integration: One can use a rule of low order of accuracy and large N or one of high order of accuracy and small N. (We venture to note that there are no extant codes for the numerical integration of systems of ordinary differential equations using Euler's rule or even Simpson's rule.) Thus, it makes no sense to compare collocation to differential inclusion assuming N is the same for each method. When collocation is used, N may thus be much smaller than when differential inclusion is used, yet the same solution accuracy achieved, and if N is smaller the number of NLP variables may be smaller for collocation, even though the control variables have been retained. When this principal (ostensible) advantage of problem size for differential inclusion is taken away, we see little remaining advantage to justify the additional work necessary to cast the problem into differential inclusion format. Three numerical examples will now be presented to illustrate this idea.

Example: Brachistochrone

The well-known brachistochrone problem of Bernoulli is one of the seminal problems in the calculus of variations. It is chosen here, for solution by both collocation and differential inclusion, because it is a nontrivial problem with an analytic solution so that the accuracy of both discrete approximate solutions to the continuous problem may be precisely determined. The problem is to find the shape of a wire so that a bead sliding on the wire without friction, in uniform gravity, will reach a given horizontal displacement in minimum time t_f. The governing equations are

$$\dot{x} = \sqrt{2gy}\cos\theta \tag{16}$$

$$\dot{y} = \sqrt{2gy}\sin\theta \tag{17}$$

with boundary conditions $x(0) = y(0) = 0$, $x(t_f)$ given. The control is the slope of the wire as a function of time $\theta(t)$.

The solution of the problem has been obtained with two collocation methods, using Simpson's rule and fifth-degree Gauss–Lobatto rule system constraints, respectively, and with the differential inclusion method. Transforming the continuous optimal control problem into a discrete, NLP problem using either collocation method is straightforward; cf. Refs. 2–4, and is not described here for brevity. Similarly, transforming the problem using differential inclusion[10] is also straightforward. Using the obvious bounds on θ of $0 \le \theta(t) \le \pi/2$ yields the following system, in which the control variable does not explicitly appear: Minimize t_f subject to

$$x_0 = 0, \qquad x_N = 0.5, \qquad y_0 = 0$$

$$\dot{\bar{x}}_i - \sqrt{2g\bar{y}_i} = \frac{x_{i+1} - x_i}{\Delta t_i} - \sqrt{2g\left(\frac{y_{i+1} + y_i}{2}\right)} \le 0$$

$$\dot{\bar{x}}_i \ge 0, \qquad \dot{\bar{y}}_i \ge 0$$

$$\dot{\bar{y}}_i - \sqrt{2g\bar{y}_i} = \frac{y_{i+1} - y_i}{\Delta t_i} - \sqrt{2g\left(\frac{y_{i+1} + y_i}{2}\right)} \le 0$$

For collocation using Simpson's rule constraints, assuming that the control is linear in time between nodal values, the number of parameters, $N_p = n \times (N+1) + m \times (N+1)$. For collocation using fifth-degree Gauss–Lobatto constraints, $N_p = n \times (2N+1) + m \times (4N+1)$. For differential inclusion, $N_p = n \times (N+1)$. Of course, for this problem $n = 2$ and $m = 1$.

Table 1 Comparison of methods used to solve the brachistochrone problem

i	Method	N	$t_{f,i}$, s	$\lvert t_{f,\text{ana}} - t_{f,i}\rvert$	N_p
	Analytic solution	——	1.2533	——	——
	Collocation				
1	Simpson	10	1.253005	0.000295	33
2	Fifth-degree Gauss–Lobatto	3	1.250521	0.002779	27
3	Fifth-degree Gauss–Lobatto	8	1.253070	0.000230	67
4	Fifth-degree Gauss–Lobatto	10	1.253183	0.000117	83
	Differential inclusion				
5	Euler	10	1.19796	0.055340	22
6	Euler	20	1.223922	0.029378	42

Table 2 Brachistochrone solution using collocation with Simpson's defects

Node i	x_i	$x_{i,\text{ana}}$	y_i	$y_{i,\text{ana}}$	θ_i, rad	$\theta_{i,\text{ana}}$
1	0.0000	0.0000	0.0000	0.0000	1.7000	1.5708
2	0.0009026	0.0008184	0.007826	0.007789	1.3330	1.4137
3	0.00706	0.006451	0.03026	0.03039	1.2788	1.2566
4	0.02157	0.02124	0.06552	0.06560	1.0936	1.0995
5	0.04903	0.04863	0.1098	0.1100	0.9442	0.9425
6	0.09116	0.09084	0.1590	0.1591	0.7851	0.7854
7	0.1489	0.1486	0.2081	0.2083	0.6286	0.6283
8	0.2214	0.2212	0.2524	0.2527	0.4713	0.4712
9	0.3066	0.3064	0.2876	0.2879	0.3142	0.3141
10	0.4009	0.4008	0.3102	0.3105	0.1571	0.1571
11	0.5000	0.5000	0.3180	0.3183	0.00007823	0.0000

The analytic solution is

$$x(t) = (gt_f/\pi)(t - (t_f/\pi)\sin\{\pi[1 - (t/t_f)]\}) \tag{18}$$

and

$$y(t) = \frac{2gt_f^2}{\pi^2}\cos^2\left[\frac{\pi}{2}\left(1 - \frac{t}{t_f}\right)\right] \tag{19}$$

which are the parametric equations of a cycloid, and the optimal control is

$$\theta(t) = (\pi/2)[1 - (t/t_f)] \tag{20}$$

For the choice $x(t_f) = 0.5$, Eq. (18) may be solved (letting $g = 1$) for the final time yielding $t_f = 1.2533$. The NLP problems resulting from the use of both collocation and differential inclusion have been solved, for this example and for the two examples to follow, using the NLP solver NZSOL,[14] an improved version of NPSOL.[15] Table 1 compares the solution methods, their approximate solutions for the final time, the error in this solution, and the number of NLP problem parameters N_p each solution requires. As expected from the analysis in the preceding section, for the same N both collocation solutions are more accurate than the corresponding differential inclusion solutions. What is perhaps unexpected is that the collocation solution using fifth-degree Gauss–Lobatto constraints, but only 3 segments (and 27 variables), is an order of magnitude more accurate than a differential inclusion solution using 20 segments (and 42 variables). The histories of the states, x and y at each node, are compared to the corresponding states from the analytic solutions for all three methods (using $N = 10$) in Tables 2–4; the collocation solutions are generally accurate to $\mathcal{O}(10^{-4})$ and the differential inclusion solution to $\mathcal{O}(10^{-2})$.

Each method was tested for robustness by running the problem with a modification of the converged solution as an initial guess. Modifications of the problem included setting all of the state and control variables to one specific number, such as one or zero, or multiplying the variables by a constant.

Both collocation methods are very robust. Convergence occurs when all of the states and controls are set to 0.1, 1.0, or 100 in the initial guess given to NZSOL.[14] Convergence also occurs when each state and control of the converged solution is multiplied by a factor of 100. Convergence does not occur if all of the states and controls are set to zero or a negative number.

Linear differential inclusion is not very robust for this problem. Convergence does not occur when all of the states are set to 0.1, 1.0, or 100 in the initial guess given to NZSOL.[14] Convergence also does not occur when each state of the converged solution is multiplied by a factor of 100. Convergence does not occur if all of the states are set to zero or a negative number. Convergence is obtained when the converged solution is modified by adding small increments such as 0.05, 0.06, and 0.07 to the state variables.

In all methods the initial guess for the final time should be a nonzero, positive number. It was chosen to be in the range of 1–5 s. It was found experimentally that to obtain convergence the initial guess at the final time should be no larger than four times the final time of the analytic solution.

Table 3 Brachistochrone solution using collocation with fifth-degree Gauss–Lobatto defects

Node i	x_i	$x_{i,\text{ana}}$	y_i	$y_{i,\text{ana}}$	θ_i, rad	$\theta_{i,\text{ana}}$
1	0.0000	0.0000	0.0000	0.0000	1.7040	1.5708
2	0.0009034	0.0008184	0.007798	0.007789	1.3312	1.4137
3	0.006631	0.006451	0.03036	0.03039	1.2793	1.2566
4	0.02141	0.02124	0.06556	0.06560	1.0935	1.0995
5	0.04879	0.04863	0.1099	0.1100	0.9443	0.9425
6	0.09098	0.09084	0.1591	0.1591	0.7851	0.7854
7	0.1487	0.1486	0.2082	0.2083	0.6826	0.6283
8	0.2213	0.2212	0.2526	0.2527	0.4713	0.4712
9	0.3065	0.3064	0.2878	0.2879	0.3142	0.3141
10	0.4008	0.4008	0.3104	0.3105	0.1571	0.1571
11	0.5000	0.5000	0.3182	0.3183	0.0000936	0.0000

Table 4 Brachistochrone solution using linear differential inclusion

Node i	x_i	$x_{i,\text{ana}}$	y_i	$y_{i,\text{ana}}$
1	0.0000	0.0000	0.0000	0.0000
2	0.001595	0.0008184	0.01417	0.007789
3	0.009413	0.006451	0.04127	0.03039
4	0.02699	0.02124	0.07890	0.06560
5	0.05699	0.04863	0.1237	0.1591
6	0.1010	0.09084	0.1717	0.1100
7	0.1593	0.1486	0.2186	0.2083
8	0.2309	0.2212	0.2603	0.2527
9	0.3138	0.3064	0.2931	0.2879
10	0.4048	0.4008	0.3140	0.3105
11	0.5000	0.5000	0.3211	0.3183

Example: Simple Cart Problem

The next example is a simple problem with linear governing equations but a quadratic performance index.[16] It also has an analytic solution, but unlike the brachistochrone it has a fixed final time. The problem can be thought of as applying a force to a cart of unit mass, when the cart is subject to drag depending linearly on the velocity, i.e., the system equations are

$$\dot{x}_1 = x_2 \tag{21}$$

$$\dot{x}_2 = -x_2 + u \tag{22}$$

The cart is initially at rest. The external force $u(t)$ is to be applied in such a way as to satisfy a terminal constraint, which is a linear combination of position and velocity, at given final time t_f

$$\psi = ax_1(t_f) + bx_2(t_f) - c = 0 \tag{23}$$

while minimizing the integral square control

$$J = \int_0^{t_f} u^2\,\mathrm{d}t \tag{24}$$

The problem has a straightforward analytic solution. If the problem constants are chosen as $a = 1.0$, $b = -2.694528$, $c = -1.155356$, and $t_f = 2.0$ the optimal control and optimal trajectory are simply

$$u(t) = \tfrac{1}{4}e^t - \tfrac{1}{2} \tag{25}$$

$$x_1(t) = -\tfrac{3}{8}e^{-t} + \tfrac{1}{8}e^t - \tfrac{1}{2}t + \tfrac{1}{4} \tag{26}$$

$$x_2(t) = \tfrac{3}{8}e^{-t} + \tfrac{1}{8}e^t - \tfrac{1}{2} \tag{27}$$

Table 5 Comparison of methods used to solve the simple cart problem

i	Method	N	J_i	$\lvert J_i - J_{\text{ana}}\rvert$	N_p
	Analytic solution	——	0.577678	——	——
	Collocation				
1	Simpson	5	0.577668	0.00001	18
2	Simpson	10	0.577678	0.00000	33
3	Simpson	20	0.577682	0.000004	63
	Differential inclusion				
4	Euler	5	0.582800	0.005122	12
5	Euler	10	0.578935	0.001257	22
6	Euler	20	0.577990	0.000312	42

Table 6 Comparison of simple cart final state and control variables

i	Method	N	$u(t_f)$	$x_1(t_f)$	$x_2(t_f)$
	Analytic solution	——	1.347264	0.122881	0.474383
	Collocation				
1	Simpson	5	1.326334	0.1227496	0.474333
2	Simpson	10	1.342595	0.122815	0.474358
3	Simpson	20	1.346748	0.122868	0.474377
	Differential inclusion				
4	Euler	5	——	0.131702	0.477656
5	Euler	10	——	0.125050	0.475188
6	Euler	20	——	0.123432	0.474587

and the objective function then becomes

$$J = \int_0^{t_f} u^2 \, \mathrm{d}t = 0.577678 \tag{28}$$

The problem has been solved with collocation (using Simpson's rule constraints) and with differential inclusion using three different discretizations for each method, i.e., $N = 5$, 10, or 20. Table 5 shows the accuracy of the objective function as a function of N. The results are consistent with the analysis in this paper; whereas differential inclusion requires fewer NLP problem variables for given choice of N, the higher order of accuracy of the Simpson's rule implicit integration used with collocation (compared to the Euler step used with differential inclusion) means that a 5-segment collocation solution using 18 total variables is orders of magnitude more accurate than even a 20-segment differential inclusion solution using 42 variables. The final values of the states are compared in Table 6 with precisely the same result. Both methods are very robust solving this problem at any discretization, for example, both converge to a solution when given initial values of 0 or -1 for all variables.

Example: F-15 Minimum Time-to-Climb

The final example in which collocation and differential inclusion will be compared is this problem, which was used as an example in Ref. 6. The problem has been stated in an earlier section; the system governing equations are Eqs. (3–5). The problem does not have an analytical solution. It has been solved here using collocation with either Simpson's rule or fifth-degree Gauss–Lobatto constraints and using differential inclusion. In the absence of an analytical solution we will assume, from the analysis in preceding sections, that the solution using the most accurate implicit quadrature rule, i.e., the solution using the fifth-degree Gauss–Lobatto rule, is the most accurate. Of course, it also employs the largest number of discrete variables (145) and correspondingly has the finest discretization in time, which further supports the assumption of its being the most accurate.

The resulting minimum final time is shown in Table 7 for the four solutions obtained. We find that the second most accurate solution is the other collocation solution, using Simpson's rule and 10 segments. The 20-segment differential inclusion solution is not only less accurate but uses more NLP problem variables. The time histories of the states are shown in Figs. 1–3 for all of the 10-segment solutions. The result in all cases is qualitatively the same as the corresponding result in Ref. 6 (cf. Fig. 3 of that paper), and so the different propulsion model used in this work has no significant effect on the solution. The three solutions differ the most in the history of the flight-path angle, and it is clear by comparing Figs. 1–3, assuming of course that the solution using the Gauss–Lobatto constraints (Fig. 3) is the most accurate, that the collocation solution does a much better job of representing the early history of the flight-path angle than the differential inclusion solution.

Table 7 Comparison of methods used to solve the F-15 problem

Number	Method	N	t_f, s	N_p
	Collocation			
1	Simpson	10	155.9904	55
2	Fifth-degree Gauss–Lobatto	10	156.2907	145
	Differential inclusion			
3	Euler	10	160.2237	33
4	Euler	20	155.8751	63

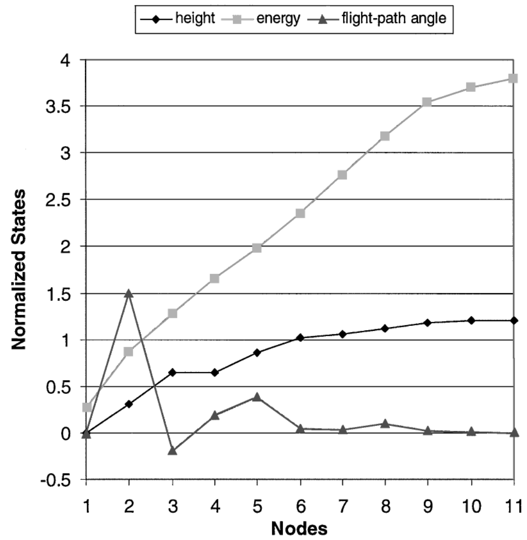

Fig. 1 Solution for discrete state time histories for F-15 minimum time-to-dash problem using differential inclusion method.

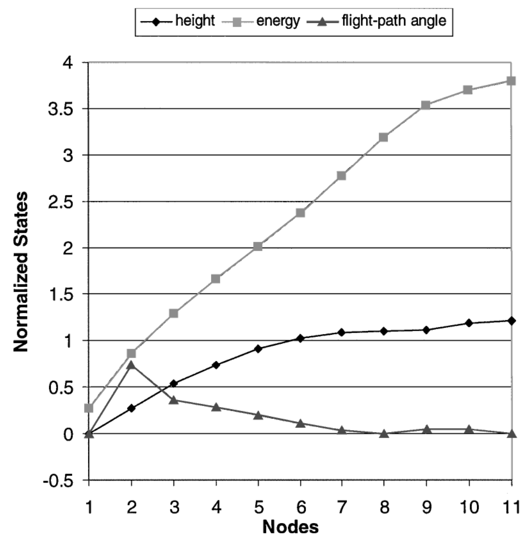

Fig. 2 Solution for discrete state time histories for F-15 minimum time-to-dash problem using collocation with Simpson method implicit integration.

The ability of each method to converge from a poor initial guess of the system variables is summarized in Table 8. For the five deliberately poor initial guesses described in Table 8, there was only one case in which the NLP problem solver would converge using a differential inclusion problem formulation but not converge using collocation. Thus, for this admittedly limited survey of

Table 8 Convergence tests using a modified F-15 solution

Modified variables	Convergence
Collocation	
$h_i, E_i, \gamma_i, \eta_i \times 1.5, n_i = 1$	Yes
$h_i = 0.8, \quad E_i = 2.0$	Yes
$E_i = 3.0$	No
$h_i, E_i, \gamma_i, \eta_i, n_i = 1$	No
$h_i, E_i, \gamma_i, \eta_i, n_i = 0$	No
Differential inclusion	
$h_i, E_i, \gamma_i \times 1.5$	Yes
$h_i = 0.8, \quad E_i = 2.0$	Yes
$E_i = 3.0$	No
$h_i, E_i, \gamma_i = 1$	No
$h_i, E_i, \gamma_i = 0$	Yes

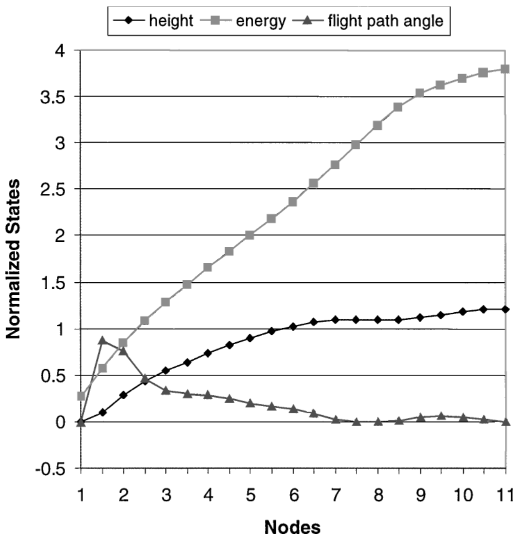

Fig. 3 Solution for discrete state time histories for F-15 minimum time-to-dash problem using collocation with fifth-degree Gauss–Lobatto implicit integration.

possible initial guesses, it may be said that collocation and differential inclusion are about equally robust.

Conclusions

All direct solutions of continuous optimal control problems require an explicit or implicit integration of the system equations of motion. The collocation method and the differential inclusion method are of the latter type; both use quadrature rules as nonlinear constraint equations. However, as the analysis here shows, the differential inclusion method requires the use of quadrature rules with very low accuracy. Collocation methods are not limited in this way and may use implicit integration rules of very high accuracy. Thus, just as is true for the qualitatively similar process of numerical integration, larger step sizes may be used, and yet the same or greater accuracy achieved. With fewer steps or segments required for a given accuracy, the NLP problem into which the continuous problem is transformed will be smaller when collocation is used, as the examples in this paper confirm.

The two methods are essentially equally robust in the example problems solved here. One can not say that one method is more robust, in general, than the other. However, it is certainly true that smaller NLP problems are easier to solve than larger ones so that whichever method yields a problem having fewer variables is more likely to lead to a converged solution. It is not impossible that a particular problem will be more tractable when cast in the differential inclusion format. Thus, one exception to the recommendation of this work, which is to retain control variables and solve direct optimization problems using collocation, would be the case in which a solution can not be found, i.e., where the NLP problem solver will not converge. Then differential inclusion might be applied as an alternative, though with no guarantee of success.

References

[1]Brauer, G., Cornick, D., and Stevenson, R., "Capabilities and Applications of the Program to Optimize Simulated Trajectories (POST)," NASA CR-2770, Feb. 1977.

[2]Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338–342.

[3]Herman, A. L., and Conway, B. A., "Direct Optimization Using Collocation Based on High-Order Gauss–Lobatto Quadrature Rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592–599.

[4]Enright, P. J., and Conway, B. A., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 994–1002.

[5]Scheel, W. A., and Conway, B. A., "Optimization of Very-Low-Thrust, Many Revolution Spacecraft Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, 1994, pp. 1185–1192.

[6]Kumar, R. R., and Seywald, H., "Should Controls Be Eliminated While Solving Optimal Control Problems via Direct Methods?," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, 1996, pp. 418–423.

[7]Betts, J. T., and Huffman, W. P., "Path-Constrained Trajectory Optimization Using Sparse Sequential Quadratic Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 1, 1993, pp. 59–68.

[8]Biegler, L. T., "From Nonlinear Programming Theory to Practical Optimization Algorithms: A Process Engineering Viewpoint," *Computers and Chemical Engineering*, Vol. 17, Oct. 1993, pp. S63–S80.

[9]Enright, P. J., and Conway, B. A., "Optimal Finite-Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 981–985.

[10]Seywald, H., "Trajectory Optimization Based on Differential Inclusion," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 3, 1994, pp. 480–487.

[11]Seywald, H., "Optimal and Suboptimal Minimum Time-To-Climb Trajectories," *Proceedings of the AIAA Guidance, Navigation, and Control Conference* (New Orleans, LA), AIAA, Washington, DC, 1994, pp. 130–136; also AIAA Paper 94-3554, Aug. 1994.

[12]Seywald, H., Cliff, E. M., and Well, K. H., "Range Optimization for a Supersonic Aircraft," *Proceedings of the AIAA Guidance, Navigation, and Control Conference* (Scottsdale, AZ), AIAA, Washington, DC, 1991, pp. 967–974; also AIAA Paper 91-2712, Aug. 1991.

[13]Seywald, H., "Long Flight-Time Range-Optimal Aircraft Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 1, 1996, pp. 242–244.

[14]Gill, P. E., Saunders, M. A., and Murray, W., "User's Guide for NZOPT 1.0: A FORTRAN Package for Nonlinear Programming," McDonnell Douglas Aerospace, Huntington Beach, CA, 1993.

[15]Gill, P. E., Saunders, M. A., and Murray, W., "User's Guide for NPSOL (4.0)," Stanford Univ., TR SOL 86-2, Stanford, CA, Jan. 1986.

[16]Barnett, S., *Introduction to Mathematical Control Theory*, Oxford Univ. Press, Oxford, England, UK, 1975, p. 224.