JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 24, No. 2, March–April 2001

New Computational Framework for Trajectory Optimization of Higher-Order Dynamic Systems

Tawiwat Veeraklaew* and Sunil K. Agrawal†
University of Delaware, Newark, Delaware 19716

By the use of tools from systems theory, it is now well known that classes of linear and nonlinear dynamic systems in first-order form can be alternatively written in higher-order form, that is, as sets of higher-order differential equations. Input-state linearization is one of the popular tools to achieve such a transformation. For mechanical systems, the equations of motion naturally have a second-order form. For real-time planning and control, a higher-order form offers a number of advantages compared to the first-order form. The question of trajectory optimization of higher-order systems with general nonlinear constraints is addressed. First, we develop the optimality conditions directly using their higher-order form. These conditions are then used to develop computational approaches. A general purpose program has been developed to benchmark computations between problems posed in alternate higher-order and first-order forms. The program implements both direct and indirect methods and uses collocation in conjunction with a nonlinear programming solver.

I. Introduction and Problem Statement

IT is conventional to consider dynamic systems in the state-space form, that is, as a set of first-order differential equations. However, the dynamics of mechanical systems has a natural second-order form, arising out of the application of Newton's laws. From a different perspective, using results from systems theory, (for example, see Ref. 1), dynamic systems can be written in canonical forms that allow the governing equations to be expressed as higher-order differential equations. Dynamic systems that have this feature include controllable linear and nonlinear systems and can be grouped under the broader umbrella of differentially flat systems.[2]

The objective of this paper is to address the problem of trajectory optimization of systems that admit both first-order and higher-order representations, either in their original coordinates or in the transformed coordinates. Methods that exploit the structure of the higher-order differential equations to compute the optimal solution efficiently are still in their infancy. However, a few studies in the literature address this issue applied to limited classes of problems. For example, a direct method was used for planar vertical takeoff and landing systems to compute the optimal solution, where the inequality constraints were not considered.[3] In recent work, the indirect method has been used to compute the optimal solution in the absence of inequality constraints. In these studies, the system equations were explicitly embedded into the cost functional. The results from higher-order variational theory were used to find the optimality conditions.[4] This approach was applied successfully to linear systems[5,6] and feedback linearizable nonlinear systems.[7]

The purpose of this paper is to extend the optimality theory applicable to higher-order systems, without converting the system equations to the first-order form. Using the resulting optimality conditions, we develop and implement direct and indirect computational algorithms. The same direct or indirect algorithm is used to solve the first-order and the higher-order optimization problems, thereby allowing a benchmark of the computation requirements in a systematic manner.

The statement of the problem is to find the optimal trajectory, that is, the function pair $[x(t), u(t)]$ for a dynamic system described by

$$x^{(p)}(t) = \boldsymbol{f}\left(x, x^{(1)}, \ldots, x^{(p-1)}, u, t\right) \tag{1}$$

where $x(t) \in \mathcal{R}^n$ and $u(t) \in \mathcal{R}^m$, $x^{(i)}(t)$ is the ith derivative of $x(t)$, and $\boldsymbol{f}$ is a vector function in $\mathcal{R}^n$. We assume that $\boldsymbol{f}$ has continuous first and second partial derivatives with respect to the arguments $x, x^{(1)}, \ldots, x^{(p-1)}$ and u. The optimal trajectory minimizes the cost

$$J = \Phi\left[x(t_f), x^{(1)}(t_f), \ldots, x^{(p-1)}(t_f), t_f\right] + \int_{t_0}^{t_f} L\left[x(\tau), x^{(1)}(\tau), \ldots, x^{(p-1)}(\tau), u(\tau), \tau\right] \mathrm{d}\tau \tag{2}$$

while satisfying the constraints

$$c_j\left(t, x, x^{(1)}, \ldots, x^{(p-1)}, u\right) \leq 0, \qquad j = 1, \ldots, r \tag{3}$$

The functions Φ, L, and c_j are also assumed to have continuous first and second partial derivatives with respect to the arguments. For $p = 1$, Eq. (1) is the familiar first-order description of the system with n states and m inputs. For $p > 1$, if Eq. (1) is written in the state-space form, it will result in np first-order differential equations with m control inputs.

The space of functions in which the extremum is sought is $\mathcal{D}_p(t_0, t_f)$ (Ref. 4). It consists of continuous functions $x(t)$ with p continuous derivatives on an interval $[t_0, t_f]$. The norm on $\mathcal{D}_p$ is defined as

$$\|x\|_p = \sum_{i=0}^{p} \max_{t_0 \leq t \leq t_f} \left|x^{(i)}(t)\right| \tag{4}$$

where $x^{(i)}(t)$ is the ith derivative of $x(t)$ and $x^{(0)}(t)$ is $x(t)$. The space $\mathcal{D}_p$ is a normed linear space.

The organization of this paper is as follows: Section II describes the optimality conditions. Section III approaches these results from the viewpoint of calculus of variations. Section IV outlines the computational algorithms and the nonlinear programming problem. The computation comparisons are made with examples in Sec. V.

II. Optimality Conditions

The optimality conditions are derived using Hamilton–Jacobi theory starting from the higher-order form of the dynamic system and the cost functional. The derivation is performed in two steps. First, it is assumed that $u(t) \in \mathcal{U}$, where the constraint set $\mathcal{U}$ is piecewise smooth and independent of x and its higher derivatives. Second, the constraint set $\mathcal{U}$ is allowed to be dependent on x and its derivatives, as required by Eq. (3). In our subsequent discussions, we use $\boldsymbol{x}(t) = [x(t)^T x^{(1)}(t)^T, \ldots, x^{(p-1)}(t)^T]^T$ to simplify the notations.

Received 29 November 1999; revision received 20 May 2000; accepted for publication 26 May 2000.

*Graduate Student, Mechanical Systems Laboratory, Department of Mechanical Engineering.

†Associate Professor, Mechanical Systems Laboratory, Department of Mechanical Engineering.

A. Constraint Set: $u \in \mathcal{U}$

Theorem: Let $\boldsymbol{x}(t)$ and $u^*(t)$ be the optimal functions that minimize Eq. (2) subject to Eqs. (1) and $u(t) \in \mathcal{U}$ starting out from an initial state $\boldsymbol{x}(t_0)$. Define a function

$$\mathcal{H}[\boldsymbol{x}(t), u(t), \lambda(t), t] = L[\boldsymbol{x}(t), u(t), t] + \lambda(t)^T f[\boldsymbol{x}(t), u(t), t] \tag{5}$$

where $\lambda(t) \in \mathcal{R}^n$ are smooth and differentiable functions that satisfy the differential equation

$$\mathcal{H}_x - \mathcal{H}^{(1)}_{x^{(1)}} + \cdots + (-1)^{p-1}\mathcal{H}^{(p-1)}_{x^{(p-1)}} = (-1)^{(p-1)}\lambda^{(p)} \tag{6}$$

with all partials evaluated along the optimal trajectory. Then, for all $t \in [t_0, t_f]$, the function $\mathcal{H}[\boldsymbol{x}(t), u^*(t), \lambda(t), t] \le \mathcal{H}[\boldsymbol{x}(t), u(t), \lambda(t), t]$, that is,

$$\mathcal{H}[\boldsymbol{x}(t), u^*(t), \lambda(t), t] = \min_{u \in \mathcal{U}} \mathcal{H}[\boldsymbol{x}(t), u, \lambda(t), t] \tag{7}$$

Proof: The proof uses the pattern of Ref. 8. We define a return function

$$J[\boldsymbol{x}(t), t, u(\tau)] = \Phi[\boldsymbol{x}(t_f), t_f] + \int_t^{t_f} L[\boldsymbol{x}(\tau), u(\tau), \tau]\,\mathrm{d}\tau \tag{8}$$

where $J[\boldsymbol{x}(t), t, u(\tau)]$ has continuous first and second partial derivatives with respect to the arguments. Here, $\boldsymbol{x}(t)$ is an admissible start point, and $u(\tau)$ is an admissible input defined over $t \le \tau \le t_f$, that is, $u \in \mathcal{U}$. For $[t, \boldsymbol{x}(t)]$, we define the minimum cost

$$J^*[\boldsymbol{x}(t), t] = \min_{u \in \mathcal{U}} \left\{ \Phi[\boldsymbol{x}(t_f), t_f] + \int_t^{t_f} L[\boldsymbol{x}(\tau), u(\tau), \tau]\,\mathrm{d}\tau \right\} \tag{9}$$

By subdividing the interval (t, t_f) into $t \le \tau \le t + \Delta t$ and $t + \Delta t < \tau \le t_f$, we can rewrite Eq. (9) as

$$J^*[\boldsymbol{x}(t), t] = \min_{u \in \mathcal{U}} \left\{ \int_t^{t+\Delta t} L\,\mathrm{d}\tau + J^*[\boldsymbol{x}(t + \Delta t), t + \Delta t] \right\} \tag{10}$$

On expanding Eq. (10) in Taylor's series about $[\boldsymbol{x}(t), t]$, we obtain

$$J^*[\boldsymbol{x}(t), t] = \min_{u \in \mathcal{U}} \left(\int_t^{t+\Delta t} L\,\mathrm{d}\tau + J^*[\boldsymbol{x}(t), t] + \frac{\partial J^*}{\partial t}[\boldsymbol{x}(t), t]\Delta t + \left\{ \frac{\partial J^*}{\partial \boldsymbol{x}}[\boldsymbol{x}(t), t] \right\}^T \dot{\boldsymbol{x}}(t)\Delta t + \text{higher-order terms} \right) \tag{11}$$

Recalling $\boldsymbol{x}(t)^T = [x(t)^T x^{(1)}(t)^T, \ldots, x^{(p-1)}(t)^T]^T$ and for small Δt, we have

$$J^*[\boldsymbol{x}(t), t] = \min_{u \in \mathcal{U}} \Big(J^*[\boldsymbol{x}(t), t] + \big\{ L[\boldsymbol{x}(t), u(t), t] + J^*_t[\boldsymbol{x}(t), t] + J^{*T}_x[\boldsymbol{x}(t), t]x^{(1)}(t) + \cdots + J^{*T}_{x^{(p-2)}}[\boldsymbol{x}(t), t]x^{(p-1)}(t) + J^{*T}_{x^{(p-1)}}[\boldsymbol{x}(t), t]f[\boldsymbol{x}(t), u(t), t] \big\}\Delta t + \mathcal{O}(\Delta t) \Big) \tag{12}$$

On neglecting higher-order terms of Δt and separating terms dependent on u, we can rewrite the preceding equation as

$$0 = J^*_t[\boldsymbol{x}(t), t] + J^{*T}_x[\boldsymbol{x}(t), t]x^{(1)}(t) + \cdots + J^{*T}_{x^{(p-2)}}[\boldsymbol{x}(t), t]x^{(p-1)}(t) + \min_{u \in \mathcal{U}} \big\{ L[\boldsymbol{x}(t), u(t), t] + J^{*T}_{x^{(p-1)}}[\boldsymbol{x}(t), t]\boldsymbol{f}[\boldsymbol{x}(t), u(t), t] \big\} \tag{13}$$

One can now define a Hamiltonian $\mathcal{H}[\boldsymbol{x}(t), u(t), J^*_{x^{(p-1)}}, t] = L[\boldsymbol{x}(t), u(t), t] + J^{*T}_{x^{(p-1)}}[\boldsymbol{x}(t), t]\boldsymbol{f}[\boldsymbol{x}(t), u(t), t]$, and

$$\mathcal{H}\big[\boldsymbol{x}(t), u^*(t), J^*_{x^{(p-1)}}, t\big] = \min_{u \in \mathcal{U}} \mathcal{H}\big[\boldsymbol{x}(t), u(t), J^*_{x^{(p-1)}}, t\big] \tag{14}$$

Here, the minimizing control is said to depend on $\boldsymbol{x}(t)$, $J^*_{x^{(p-1)}}[\boldsymbol{x}(t), t]$, and t. From Eqs. (13) and (14), the extended form of Hamilton–Jacobi equation is

$$J^*_t[\boldsymbol{x}(t), t] + J^{*T}_x[\boldsymbol{x}(t), t]x^{(1)}(t) + \cdots + J^{*T}_{x^{(p-2)}}[\boldsymbol{x}(t), t]x^{(p-1)}(t) + \mathcal{H}\big\{\boldsymbol{x}(t), u^*\big[\boldsymbol{x}(t), J^*_{x^{(p-1)}}, t\big], J^*_{x^{(p-1)}}, t\big\} = 0 \tag{15}$$

where J^* satisfies the boundary condition

$$J^*[\boldsymbol{x}(t_f), t_f] = \Phi[\boldsymbol{x}(t_f), t_f] \tag{16}$$

In summary, 1) the optimal control $u^*[\boldsymbol{x}(t), J^*_{x^{(p-1)}}, t]$ minimizes the Hamiltonian $\mathcal{H}$ defined in Eq. (14) and 2) the optimal return function satisfies the partial differential equation (15). If $p = 1$, these results simplify to the classical Hamilton–Jacobi equations.[9]

The costate equations (6) can be derived from Hamilton–Jacobi Eqs. (15) using the procedure suggested by Kirk.[8] If $[\boldsymbol{x}^*(t), t]$ is a point on the optimal trajectory, the Hamilton–Jacobi equation can also be written as

$$0 = \min_{u \in \mathcal{U}} \Big\{ J^*_t[\boldsymbol{x}^*(t), t] + J^{*T}_x[\boldsymbol{x}^*(t), t]x^{*(1)}(t) + \cdots + J^{*T}_{x^{(p-2)}}[\boldsymbol{x}^*(t), t]x^{*(p-1)}(t) + \mathcal{H}\big[\boldsymbol{x}^*(t), u(t), J^*_{x^{(p-1)}}, t\big] \Big\} \tag{17}$$

since $J^*_t[\boldsymbol{x}^*(t), t]$, $J^{*T}_x[\boldsymbol{x}^*(t), t]\ x^{*(1)}(t), \ldots, J^{*T}_{x^{(p-2)}}\ [\boldsymbol{x}^*(t), t]\ x^{*(p-1)}(t)$ are independent of $u(t)$. In words, for a $[\boldsymbol{x}^*(t), t]$, the control $u^*(t)$ minimizes the right-hand side of Eq. (17), and the minimum is zero. Hence, if we define a function

$$v[\boldsymbol{x}(t), u^*(t), t] = J^*_t[\boldsymbol{x}(t), t] + J^{*T}_x[\boldsymbol{x}(t), t]x^{(1)}(t) + \cdots + J^{*T}_{x^{(p-2)}}[\boldsymbol{x}(t), t]x^{(p-1)}(t) + \mathcal{H}\big[\boldsymbol{x}(t), u^*(t), J^*_{x^{(p-1)}}, t\big] \tag{18}$$

in the neighborhood of $\boldsymbol{x}^*(t)$, that is, $\boldsymbol{x}(t) = \boldsymbol{x}^*(t) + \delta\boldsymbol{x}(t)$, this function has a local mimimum at $\boldsymbol{x}^*(t)$, that is, $\partial v/\partial \boldsymbol{x}[\boldsymbol{x}^*(t), u^*(t), t] = 0$. Because the mixed partial derivatives are continuous, the order of the derivatives in a mixed partial can be interchanged. With this property, $\partial v/\partial \boldsymbol{x}[\boldsymbol{x}^*(t), u^*(t), t] = 0$ simplifies to the following component equations:

$$J^*_{x^{(k)}t} + J^*_{x^{(k)}x}x^{(1)} + \cdots + J^*_{x^{(k)}x^{(k-1)}}x^{(k)} + J^*_{x^{(k-1)}} + \cdots + J^*_{x^{(k)}x^{(p-2)}}x^{(p-1)} + J^*_{x^{(k)}x^{(p-1)}}x^{(p)} + L_{x^{(k)}} + f^T_{x^{(k)}}J^*_{x^{(p-1)}} = 0, \qquad k = 0, \ldots, p-1 \tag{19}$$

evaluated at $\boldsymbol{x}^*(t)$, $u^*(t)$, and t. Here, a term such as $J^*_{x^{(k)}x^{(p-2)}}$ is an $(n \times n)$ matrix with rs element $J^*_{x_r^{(k)}x_s^{(p-2)}}$. By the use of the definition of total time derivative of function $J^*_{x^{(k)}}$, the preceding equation simplifies to

$$\frac{\mathrm{d}J^*_{x^{(k)}}}{\mathrm{d}t} + J^*_{x^{(k-1)}} + L_{x^{(k)}} + f^T_{x^{(k)}}J^*_{x^{(p-1)}} = 0, \qquad k = 0, \ldots, p-1 \tag{20}$$

On defining $\psi_k(t) = J^*_{x^{(k-1)}}[\boldsymbol{x}^*(t), t]$, we can write Eq. (20) as

$$\psi^{(1)}_{k+1}(t) + \psi_k(t) + \mathcal{H}_{x^{(k)}}[\boldsymbol{x}^*(t), u^*(t), t] = 0, \qquad k = 0, \ldots, p-1 \tag{21}$$

Note that ψ_k are defined for values 1–p. In summary, each $[\boldsymbol{x}^*(t), t]$ on the optimal path satisfies Eq. (21). The components of this equation are

$$\begin{aligned}
&\psi^{(1)}_p(t) + \psi_{p-1}(t) + \mathcal{H}_{x^{(p-1)}}[\boldsymbol{x}^*(t), u^*(t), t] = 0\\
&\psi^{(1)}_{p-1}(t) + \psi_{p-2}(t) + \mathcal{H}_{x^{(p-2)}}[\boldsymbol{x}^*(t), u^*(t), t] = 0\\
&\vdots\\
&\psi^{(1)}_2(t) + \psi_1(t) + \mathcal{H}_{x^{(1)}}[\boldsymbol{x}^*(t), u^*(t), t] = 0\\
&\psi^{(1)}_1(t) + \mathcal{H}_x[\boldsymbol{x}^*(t), u^*(t), t] = 0
\end{aligned} \tag{22}$$

By the use of these equations, it is possible to eliminate $\psi_{p-1}(t)$–$\psi_1(t)$. Then, the resulting differential equation is

$$\mathcal{H}_x - \mathcal{H}^{(1)}_{x^{(1)}} + \cdots + (-1)^{p-1}\mathcal{H}^{(p-1)}_{x^{(p-1)}} = (-1)^p \psi_p^{(p)} \tag{23}$$

which must hold at each point $[\boldsymbol{x}^*(t), t]$ of the optimal solution. When we compare Eqs. (6) and (23), it is clear that $\lambda(t)$ are same as $\psi_p(t)$. □

The transition from Eq. (22) to Eq. (23) presupposes the existence of $\mathcal{H}^{(1)}_{x^{(1)}}, \mathcal{H}^{(2)}_{x^{(2)}}, \ldots, \mathcal{H}^{(p)}_{x^{(p)}}$. This additional hypothesis is not needed when deriving through calculus of variations. This will be discussed in more detail in Sec. III. The solution of this differential equation requires boundary conditions $\psi_p(t_f), \ldots, \psi_p^{(p-1)}(t_f)$, which can be obtained from Eqs. (16) and (22). When $p = 1$, these results simplify to the well-known optimality conditions.[10]

B. Constraint Set: $u \in \mathcal{A}(t, x)$

In this section, we consider the constraint set for $u(t)$ both time and state dependent, that is $u \in \mathcal{A}(t, \boldsymbol{x})$. In the derivation of Hamilton–Jacobi equations, the steps are same as the case $u \in \mathcal{U}$ with Eq. (14) replaced in the following way:

$$\mathcal{H}\left\{\boldsymbol{x}(t), u^*\left[\boldsymbol{x}(t), J^*_{x^{(p-1)}}, t\right], J^*_{x^{(p-1)}}, t\right\} = \min_{u \in \mathcal{A}(t,\boldsymbol{x})} \mathcal{H}\left[\boldsymbol{x}(t), u(t), J^*_{x^{(p-1)}}, t\right] \tag{24}$$

In the derivation of the costate equations, the steps remain essentially the same. Now $\boldsymbol{x}^*(t)$ minimizes $v[\boldsymbol{x}(t), u^*(t), t]$ subject to the constraint $u(t) \in \mathcal{A}(t, \boldsymbol{x})$, described as

$$C_j(t, \boldsymbol{x}, u) \leq 0, \qquad j = 1, \ldots, r \tag{25}$$

Hence, $\boldsymbol{x}^*(t)$ minimizes $v'[\boldsymbol{x}(t), u^*(t), t]$, where

$$v'[\boldsymbol{x}(t), u^*(t), t] = v[\boldsymbol{x}(t), u^*(t), t] + \sum_{j=1}^{r} \mu_j\left[C_j(t, \boldsymbol{x}, u^*) + \xi_j^2\right] \tag{26}$$

This local minimum property can be characterized as $\partial v'/\partial \boldsymbol{x}[\boldsymbol{x}^*(t), u^*(t), t] = 0$. On defining a modified Hamiltonian

$$\mathcal{H}' = \mathcal{H} + \sum_{j=1}^{r} \mu_j C_j(t, \boldsymbol{x}, u^*)$$

one can show that

$$\psi^{(1)}_{k+1}(t) + \psi_k(t) + \mathcal{H}'_{x^{(k)}}[\boldsymbol{x}^*(t), u^*(t), t] = 0 \qquad k = 0, \ldots, p-1 \tag{27}$$

and $\mu_j \xi_j = \mu_j C_j(t, \boldsymbol{x}^*, u^*) = 0$, $j = 1, \ldots, r$. This second condition says that if the jth constraint is active, $\mu_j > 0$, otherwise $\mu_j = 0$. The costate Eqs. (23) now get modified to

$$\mathcal{H}'_x - \mathcal{H}'^{(1)}_{x^{(1)}} + \cdots + (-1)^{p-1}\mathcal{H}'^{(p-1)}_{x^{(p-1)}} = (-1)^p \psi_p^{(p)} \tag{28}$$

which must hold at each point $[\boldsymbol{x}^*(t), t]$ of the optimal solution.

In summary, the optimal trajectory for a higher-order system with constraints $u \in \mathcal{A}(t, \boldsymbol{x})$ satisfies Eqs. (1), (24), and (28) along with the condition $\mu_j > 0$ if jth constraint is active, otherwise $\mu_j = 0$.

III. Variational Theory

The objective of this section is to derive optimality conditions for the problem posed in the Introduction using results from variational theory. Note that in modern literature, almost all treatments of optimal control problems consider functionals dependent on only the first derivatives.[10–13] Because our problem involves higher derivatives, we first review some general results from variational theory applicable to functionals with higher derivatives. These results are then specialized to the problem defined in the Introduction.

A. Key Results and First Integrals

We choose a general functional of the form

$$J = \Psi\left[x(t_f), x^{(1)}(t_f), \ldots, x^{(p-1)}(t_f), t_f\right] + \int_{t_0}^{t_f} F\left[x(t), x^{(1)}(t), \ldots, x^{(p-1)}(t), x^p(t), t\right] \mathrm{d}t$$

and consider the general open end states and end time problem. The problem can be stated as follows: Among all functions $x(t)$ belonging to $\mathcal{D}_p(t_0, t_f)$ with open end states and end time, find the function(s) that extremize the functional J. We replace $x(t)$ by $x(t) + h(t)$, where $h(t)$, like $x(t)$, belongs to $\mathcal{D}_p(t_0, t_f)$. By the variation δJ, we mean the expression that is linear in $h(t)$, $h^{(1)}(t)$, and $h^{(p)}(t)$ and that differs from the increment $\triangle J = J[x + h] - J[x]$ by a quantity of order higher than 1 relative to $h(t)$, $h^{(1)}(t)$, $\ldots, h^{(p)}(t)$.

From the principles of variational calculus,[4,14] one can show that

$$\delta J = \Psi_t \delta t|_{t_f} + \left[h^T \Psi_x + h^{(1)T}\Psi_{x^{(1)}} + \cdots + h^{(p-1)T}\Psi_{x^{(p-1)}}\right]_{t_f} + \int_{t_0}^{t_f}\left(h^T F_x + h^{(1)^T} F_{x^{(1)}} + \cdots + h^{(p)^T} F_{x^{(p)}}\right)\mathrm{d}t \tag{29}$$

On repeatedly integrating Eq. (29) by parts and using the boundary conditions, one can show that

$$\begin{aligned}\delta J = {} & \int_{t_0}^{t_f} h^T\left[F_x - F^{(1)}_{x^{(1)}} + \cdots + (-1)^p F^{(p)}_{x^{(p)}}\right]\mathrm{d}t \\ & + \left[h^T\left(F_{x^{(1)}} - F^{(1)}_{x^{(2)}} + \cdots + (-1)^{p-1} F^{(p-1)}_{x^{(p)}}\right)\right]_{t_0}^{t_f} \\ & + \left\{h^{(1)T}\left[F_{x^{(2)}} - F^{(1)}_{x^{(3)}} + \cdots + (-1)^{p-2} F^{(p-2)}_{x^{(p)}}\right]\right\}_{t_0}^{t_f} + \cdots \\ & + \left[h^{(p-1)T} F_{x^{(p)}}\right]_{t_0}^{t_f} + \Big[F - x^{(1)T}\left\{F_{x^{(1)}} - F^{(1)}_{x^{(2)}} + \cdots \right. \\ & \left. + (-1)^{p-1} F^{(p-1)}_{x^{(p)}}\right\} - x^{(2)T}\left\{F_{x^{(2)}} - F^{(1)}_{x^{(3)}} + \cdots \right. \\ & \left. + (-1)^{p-2} F^{(p-2)}_{x^{(p)}}\right\} - \cdots - x^{(p)T} F_{x^{(p)}}\Big]\delta t\Big|_{t_0}^{t_f} + \Psi_t \delta t\Big|_{t_f} \\ & + \left[h^T \Psi_x + h^{(1)T}\Psi_{x^{(1)}} + \cdots + h^{(p-1)T}\Psi_{x^{(p-1)}}\right]_{t_0}^{t_f}\end{aligned} \tag{30}$$

where $h^{(i)} = \delta x^{(i)}$ is the variation of ith derivative of x and δt is the variation of time. The necessary conditions are derived by making $\delta J = 0$ and consist of a set of differential equations to be satisfied by the problem along with appropriate boundary conditions. From Eq. (30), the governing differential equation is obtained from the terms within the integral

$$F_x - F^{(1)}_{x^{(1)}} + \cdots + (-1)^p F^{(p)}_{x^{(p)}} = 0 \tag{31}$$

In general, this differential equation is a $2p$th-order differential equation. This equation is the extended Euler–Lagrange equation and for $p = 1$ reduces to the more familiar costate equations in the optimal control literature.

The derivation of Eq. (31) has not been rigorously done here because the transition from Eq. (29) to Eq. (30) presupposes the existence of the derivatives $F^{(1)}_{x^{(1)}}, \ldots, F^{(p)}_{x^{(p)}}$. However, by some more elaborate arguments,[4] it can be shown that Eq. (29) implies Eq. (30), without this additional hypothesis. In fact, the arguments in question prove the existence of these derivatives. Furthermore, any $x(t)$ that has continuous p derivatives and satisfies Euler–Lagrange Eqs. (31) possesses continuous $2p$ derivatives.

The differential Eqs. (31) admits a number of first integrals depending on the structure of the integrand F.

1) If F does not explicitly contain x, that is, $F_x = 0$, Eq. (31) becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[F_{x^{(1)}} - F^{(1)}_{x^{(2)}} + \cdots + (-1)^{p-1} F^{(p-1)}_{x^{(p)}}\right] = 0 \tag{32}$$

Hence, the optimal solution admits n first integrals $F_{x^{(1)}} - F^{(1)}_{x^{(2)}} + \cdots + (-1)^{p-1} F^{(p-1)}_{x^{(p)}} = K$. With a similar reasoning, if F does not explicitly contain an element of x, for example, x_i, correspondingly, there is a single first integral.

2) If F does not explicitly depend on x and $x^{(1)}$, one can write Eq. (31) as

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\left[F_{x^{(2)}} - F^{(1)}_{x^{(3)}} + \cdots + (-1)^{p-2} F^{(p-1)}_{x^{(p)}}\right] = 0 \tag{33}$$

Hence, the optimal solution admits the integral $F_{x^{(2)}} - F^{(1)}_{x^{(3)}} + \cdots + (-1)^{p-2} F^{(p-1)}_{x^{(p)}} = K$. This argument can be extended to obtain other first integrals depending on the elements of x and their higher derivatives in the integral.

3) If F does not explicitly contain t,

$$F - x^{(1)T}\left\{F_{x^{(1)}} - F^{(1)}_{x^{(2)}} + \cdots + (-1)^{p-1} F^{(p-1)}_{x^{(p)}}\right\} - x^{(2)T}\left\{F_{x^{(2)}} - F^{(1)}_{x^{(3)}} + \cdots + (-1)^{p-2} F^{(p-2)}_{x^{(p)}}\right\} - \cdots - x^{(p)T} F_{x^{(p)}} = K \tag{34}$$

This property can be verified by time derivation of the left-hand and the right-hand sides. For open end time problems, where Ψ does not explicitly depend on t, that is, $\Psi_t = 0$, the constant $K = 0$. For $p = 1$, if F is independent of time, we arrive at the familiar result $F - x^{(1)T} F_{x^{(1)}} = K$.

B. Higher-Order Systems and First Integrals

To address the dynamic optimization problem at hand, which consists of minimizing Eq. (2) subject to Eqs. (1) and (3), one can define F and Ψ of Eq. (30) in the following way:

$$F\left(x, x^{(1)}, \ldots, x^{(p-1)}, x^{(p)}, u, \lambda, \xi, S, t\right) = L + \lambda^T\left(\boldsymbol{f} - x^{(p)}\right) + \xi^T(C + \boldsymbol{S}^2) \tag{35}$$

$$\Psi\left[x(t_f), x^{(1)}(t_f), \ldots, x^{(p-1)}(t_f), t_f\right] = \Phi \tag{36}$$

where $\boldsymbol{S}^2(t) = [s_1^2(t)\, s_2^2(t), \ldots, s_r^2(t)]^T$ is a $(r \times 1)$ positive slack vector for the constraints C, also $\xi(t)$ are the corresponding Lagrange multipliers and $\lambda(t)$ are Lagrange multipliers corresponding to the dynamic equations. By the use of the general result of Eq. (30) and the recognition that F has dependence on other variables besides x and its higher derivatives, it is immediately clear that the optimal solution must satisfy the following differential equations:

$$F_x - F^{(1)}_{x^{(1)}} + \cdots + (-1)^p F^{(p)}_{x^{(p)}} = 0 \tag{37}$$

$$F_u = 0 \tag{38}$$

$$\xi_i s_i = 0, \qquad i = 1, \ldots, r \tag{39}$$

along with Eqs. (1) and (25). Using an expression for $\mathcal{H}' = L + \lambda^T f + \xi^T C$, one can easily show that Eq. (37) simplifies to

$$\mathcal{H}'_x - \mathcal{H}'^{(1)}_{x^{(1)}} + \cdots + (-1)^{p-1} \mathcal{H}'^{(p-1)}_{x^{(p-1)}} = (-1)^p \lambda^{(p)} \tag{40}$$

and has a structure identical to Eq. (28). Equation (38) can be evaluated to show that

$$F_u = \mathcal{H}'_u = 0 \tag{41}$$

One interpretation of Eq. (41) is that, on the optimal trajectory, $u(t)$ pointwise minimizes $\mathcal{H}'$. An equivalent statement is that $u(t)$ pointwise minimizes $\mathcal{H}$ subject to the constraints (25). Hence, $u^*(t)$ of Eq. (41) also satisfies the following property:

$$\mathcal{H}[\boldsymbol{x}(t), u^*(t), \lambda, t] = \min_{u \in \mathcal{A}(t, \boldsymbol{x})} \mathcal{H}[\boldsymbol{x}(t), u(t), \lambda, t] \tag{42}$$

identical to Eq. (24). Equation (39) has the following interpretation: if $C_i < 0$, then $\xi_i = 0$ and if $C_i = 0$, then $\xi_i > 0$.

Equation (40) requires boundary conditions on higher derivatives of λ. These boundary conditions can be obtained from Eq. (30). Because initial conditions on $x(t), x^{(1)}(t), \ldots, x^{(p-1)}(t)$ are specified, $h(t_0), h^{(1)}(t_0), \ldots, h^{(p-1)}(t_0)$ are zero. Because terminal conditions on these variables are free, the terms associated with $h(t_f), h^{(1)}(t_f), \ldots, h^{(p-1)}(t_f)$ must be identically zero. This reasoning gives us the following boundary conditions:

$$\left[\mathcal{H}'_{x^{(k+1)}} - \mathcal{H}'^{(1)}_{x^{(k+2)}} + \cdots + (-1)^{p-k-2} \mathcal{H}'^{(p-k-2)}_{x^{(p-1)}} - (-1)^{p-k-1} \lambda^{(p-k-1)} + \Phi_{x^{(k)}}\right]_{t_f} = 0, \qquad k = 0, \ldots, p-1 \tag{43}$$

In summary, the functions ψ_p defined in Sec. II are essentially the same as λ introduced in this section. Both satisfy the same differential equations (28) or (40) and boundary conditions. The optimal $u(t)$ from both approaches are the same.

For the system under consideration, $F = \mathcal{H}' - \lambda^T x^{(p)}$ and $\mathcal{H}' = L + \lambda^T \boldsymbol{f} + \xi^T C$. If $L, \boldsymbol{f}$, and C are not explicit functions of time, according to the preceding section, the solution has a first integral given by Eq. (34). On simplifying, it can be shown that this equation reduces to

$$\mathcal{H}' - x^{(1)^T}\left[\mathcal{H}'_{x^{(1)}} - \mathcal{H}'^{(1)}_{x^{(2)}} + \cdots + (-1)^{p-2} \mathcal{H}'^{(p-2)}_{x^{(p-1)}} + (-1)^p \lambda^{(p-1)}\right] - x^{(2)^T}\left[\mathcal{H}'_{x^{(2)}} - \mathcal{H}'^{(1)}_{x^{(3)}} + \cdots + (-1)^{p-3} \mathcal{H}'^{(p-3)}_{x^{(p-1)}} + (-1)^{p-1} \lambda^{(p-2)}\right] - \cdots - x^{(p-1)^T}\left[\mathcal{H}'_{x^{(p-1)}} + \lambda^{(1)}\right] = K \tag{44}$$

IV. Numerical Solution via Nonlinear Programming

A nonlinear programming problem minimizes $J(y)$ subject to constraints $c(y) \leq 0$, where y are the decision variables. The optimal control problem outlined in this paper can be transformed to a nonlinear programming problem by finite parameterization of the dynamic variables. In this section, we provide details of a direct and an indirect scheme to solve the dynamic optimization problem. Both schemes parameterize the dynamic variables followed by collocation over the time domain of interest.

A. Direct Scheme and Computational Issues

In the direct scheme, the cost in Eq. (2) is optimized subject to Eqs. (1), (3), and the boundary conditions specified in the problem. In this scheme, the total time is divided into a finite number of intervals. In each interval, the state variables are represented by polynomials, with degree larger than or equal to $p - 1$. The control inputs are selected to be a constant in an interval. The differential equations (1) and the inequality constraints (3) are then satisfied at collocation points chosen over t_0 and t_f. Similar methods have been used in direct schemes for system equations in first-order forms.[15,16]

With N intervals and the parameterization of x and u, the total number of decision variables y is $\geq (np + m)N$. The variables x are subject to boundary conditions at t_0 and/or t_f and continuity across the interval boundaries. This results in a maximum of $np(N + 1)$ equality constraints. If N_c is the number of collocation points, Eq. (1) results in nN_c equality constraints and Eq. (3) in rN_c inequality constraints. The nonlinear programming problem is now well defined and can be solved by standard available tools.

Here, we briefly compare the dimensions of two nonlinear programs applicable to the same physical problem, one posed in a higher-order form and the other in an alternate first-order form. We assume the same number of intervals and collocation points in the two problems.

A system described by n pth-order differential equations is also equivalent to a system with np first-order differential equations. For the first-order problem, the parameterization of the states in each interval must be at least first-degree polynomials, and the inputs are piecewise constants. This results in the total number of decision variables to be larger than or equal to $(np + m)N$. The boundary conditions and continuity across the nodes add up to a maximum of $np(N + 1)$ equality constraints. If N_c is the number of collocation points, the dynamic equations in the first order will yield npN_c equality constraints and rN_c inequality constraints.

On comparing these numbers with those obtained for the higher-order problem, one can make the following observation: For the same collocation grid in time, the dynamic equations in the higher-order form yield equality constraints that are smaller by a factor p compared to those obtained for the first-order form.

B. Indirect Scheme

The indirect computation scheme is similar to the direct scheme, except that the first-order optimality equations are explicitly considered. Besides Eqs. (1–3), the additional equations considered are Eqs.(40–42). These are n state and n costate equations, r inequality constraints, and m scalar conditions $\mathcal{H}'_u = 0$. The dynamic variables to be determined are n state variables, n costate variables, m control inputs, and r Lagrange variables corresponding to the constraints.

The total time is divided into a finite number of intervals. In each interval, the state variables, the costate variables, and Lagrange variables are represented by piecewise polynomials, with degrees larger than or equal to $2p-1$. At each interval boundary, $2p$ continuity conditions are imposed at the nodes. These are 1) continuity of $x(t)$ up to $p-1$ derivatives across a node and 2) continuity of the natural boundary term across a node

$$\left[\mathcal{H}'_{x^{(i)}} - \mathcal{H}'^{(1)}_{x^{(i+1)}} + \cdots + (-1)^{p-3}\mathcal{H}'^{(p-2-i)}_{x^{(p-1)}} + (-1)^{p-i}\lambda^{(p-i)}\right]_{t_i^-}$$
$$= \left[\mathcal{H}'_{x^{(i)}} - \mathcal{H}'^{(1)}_{x^{(i+1)}} + \cdots + (-1)^{p-3}\mathcal{H}'^{(p-2-i)}_{x^{(p-1)}} + (-1)^{p-i}\lambda^{(p-i)}\right]_{t_i^+}, \qquad i = 1, \ldots, p \tag{45}$$

The control inputs are selected to be piecewise constants. Equations (1–3) and Eqs. (40–42) are then satisfied at a finite collocation grid over t_0 and t_f. Similar methods have been used in indirect optimization of problems posed in first-order forms.[17]

With N intervals and the parameterization of x, λ, u, and μ, the total number of decision variables y is $\geq(3np+m)N$. The variables x are subject to boundary conditions at t_0 and/or t_f and continuity across the interval boundaries. This results in a maximum of $2np(N+1)$ equality constraints. If N_c is the number of collocation points, Eqs. (1) and (40) each result in nN_c equality constraints. The control optimality equations $\mathcal{H}'_u = 0$ yield mN_c equality constraints and Eq. (3) yields rN_c inequality constraints. The nonlinear programming problem is now well defined and can be solved by standard available tools.

Because the state and the costate equations can be alternatively written in first-order forms, one can reason that, for each collocation point, the number of inequality constraints in the first-order and higher-order forms are the same. The substantial difference in equality constraints comes from the two alternative forms of the dynamic equations and the costate equations. Similar to the direct case, one can make the following observation: For the same collocation grid in time, the dynamic equations and the costate equations in the higher-order form yield equality constraints that are smaller by a factor p compared to those obtained for the first-order form.

C. General Program

A general purpose program was developed that implements direct and indirect schemes for a system described in higher-order form. The first-order form can be treated as a special case with $p=1$. This program is written in MATLAB® with an interface to a nonliner programming solver NPSOL. All of the steps leading to the optimality equations are performed in Symbolic MATLAB. The procedures for parameterizing the dynamic variables and collocation with finite grids are coded in MATLAB. The analytical gradients of the resulting constraints are passed on to NPSOL.

V. Numerical Examples

The methods proposed are illustrated by two examples. The first example is a linear spring–mass–damper system, whereas the second is a nonlinear system. Starting from Newton's laws, both examples can be written in second-order forms. Alternatively, they can be written in first-order forms as well as fourth-order forms, as described later.

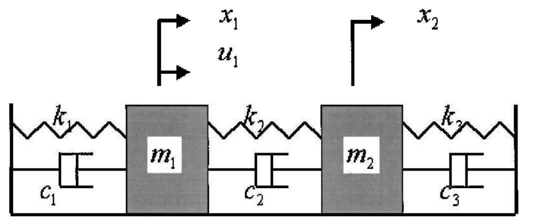


Fig. 1 Spring–mass–damper system with two masses and one input.

A. Spring–Mass–Damper System

This system is shown in Fig. 1. The equations of motion are

$$Mx^{(2)} + Cx^{(1)} + Kx = \begin{bmatrix} u \\ 0 \end{bmatrix} \tag{46}$$

where, $x = (x_1 \;\; x_2)^T$. The matrices M, C, and K are

$$M = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}, \qquad C = \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 + c_3 \end{bmatrix}$$
$$K = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix} \tag{47}$$

In the numerical solution, the parameters in meter–kilogram–second (MKS) units are chosen as $m_1 = m_2 = 1.0$, $c_1 = c_3 = 1.0$, $c_2 = 2.0$, and $k_1 = k_2 = k_3 = 3.0$. The objective is to minimize the cost

$$J = \int_{t_0}^{t_f} u^2 \,\mathrm{d}t$$

and steer the system from $x(t_0) = (10 \;\; 20)^T$ to $x(t_f) = (10 \;\; 20)^T$ while starting and reaching at rest. The input must satisfy the constraint $-300 \leq u \leq 300$ during motion. The final time t_f is 2.0 s.

A first-order form for this system is

$$\bar{q}^{(1)} = \bar{A}\bar{q} + \bar{B}u \tag{48}$$

where

$$\bar{A} = \begin{bmatrix} 0 & I_n \\ -M^{-1}K & -M^{-1}C \end{bmatrix}, \qquad \bar{B} = \begin{bmatrix} 0 \\ 0 \\ 1/m_1 \\ 0 \end{bmatrix}, \qquad \bar{q} = \begin{bmatrix} x \\ x^{(1)} \end{bmatrix} \tag{49}$$

The fourth-order form can be obtained by invoking a state transformation of the form $\bar{q} = Tq$ to Eq. (48), where T is the controllability matrix corresponding to the pair $(\bar{A}, \bar{B})$. In the space of transformed variables $q = (q_1 \;\; q_2 \;\; q_3 \;\; q_4)^T$, the dynamic equations have a fourth-order form.[5] For the given parameters of the problem, this equation is

$$q_4^{(4)} + 6q_4^{(3)} + 17q_4^{(2)} + 24q_4^{(1)} + 27q_4 = u \tag{50}$$

where q_4 and u are scalars.

From Eq. (50), the expression for $u(t)$ can also be substituted in the cost functional J and inequality constraints. Hence, an alternate representation of the optimization problem is to find a trajectory of $q_4(t)$ that minimizes the cost functional

$$J = \int_{t_0}^{t_f} \left(q_4^{(4)} + 6q_4^{(3)} + 17q_4^{(2)} + 24q_4^{(1)} + 27q_4\right)^2 \mathrm{d}t \tag{51}$$

and satisfies the constraint

$$-300 \leq q_4^{(4)} + 6q_4^{(3)} + 17q_4^{(2)} + 24q_4^{(1)} + 27q_4 \leq 300 \tag{52}$$

In summary, there are four alternative descriptions of the same problem: 1) the second-order form of Eq. (47) with two variables and one input, 2) the first-order form of Eq. (49) with four variables and one input, 3) the fourth-order form of Eq. (50) with one variable and

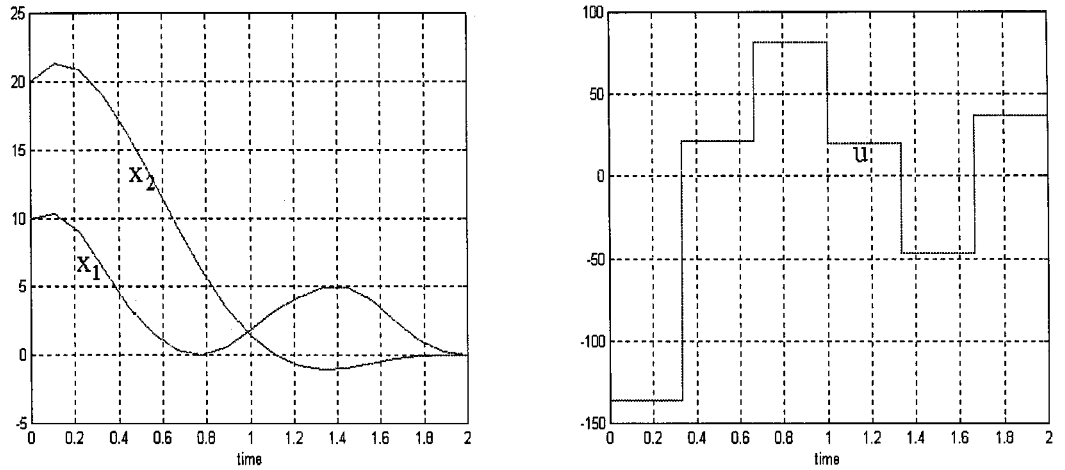


Fig. 2 Six-interval direct solution of x_1, x_2, and u for example 1; plots for the four cases 1–4 overlap within the accuracy of the figure.

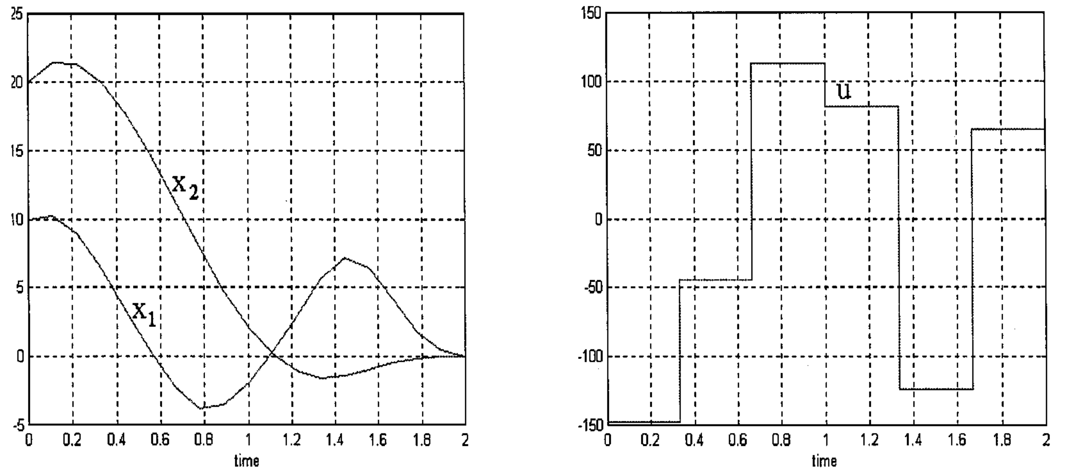


Fig. 3 Six-interval indirect solution of x_1, x_2, and u for example 1; plots for the four cases 1–4 overlap within the accuracy of the figure.

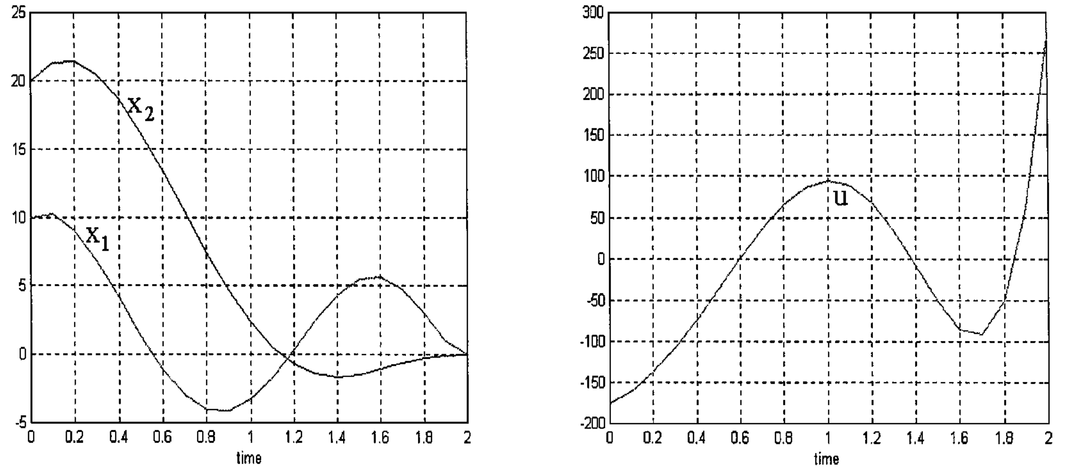


Fig. 4 Analytical solution for x_1, x_2, and u using matrix exponential.

one input, and 4) the fourth-order form with the dynamic equations eliminated from the problem.

The six-interval direct solution for this example is shown in Fig. 2. The plots for the four cases overlap within the accuracy of Fig. 2. The six-interval indirect solution for this example is shown in Fig. 3 for the four cases, which again overlap within the accuracy of the figure. Because the solution does not hit any constraints, the analytical solution for x_1, x_2, and u using matrix exponential is shown in Fig. 4.

In this problem, the time does not appear explicitly. As described in Sec. III, the solution has a first integral. This first integral was also explicitly used to refine the indirect solution to yield results closest to the closed-form solution. Hence, the solution where the first integral was explicitly used as a constraint, was taken as a benchmark to compare the accuracy of solutions obtained from the various approaches. Figure 5 shows the plots comparing the accuracy of the solutions.

Table 1 shows the CPU run-time for direct/indirect and first-order/higher-order solution schemes. Some salient points that we observe from Table 1 are as follows: 1) A fourth-order direct or indirect solution has roughly an order of magnitude smaller CPU time compared

Table 1 CPU run-time of example 1 showing direct/indirect and first-order/higher-order comparisons

Systems	CPU run-time, s
Direct first order	20.66
Direct second order	7.800
Direct fourth order	2.91
Direct fourth order eliminating dynamic equations	7.03
Indirect first order	99.74
Indirect second order	22.35
Indirect fourth order	15.87
Indirect fourth order eliminating dynamic equations	20.83

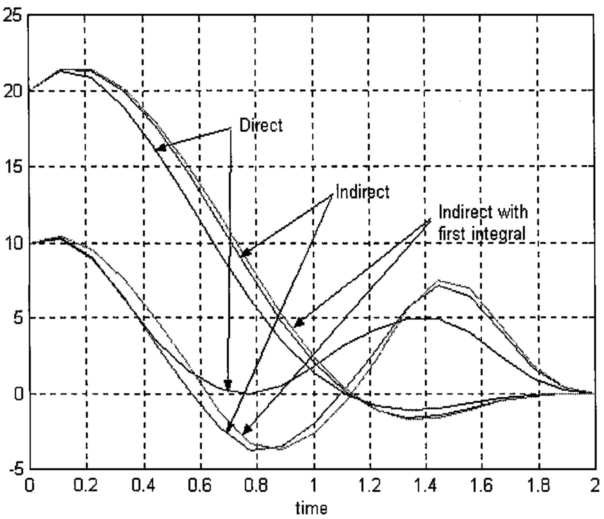
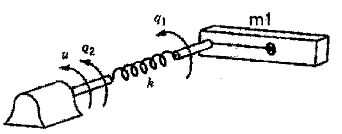


Fig. 5 Accuracy of the solutions in comparison with the solution obtained with the first integral used explicitly as a constraint in example 1.

Fig. 6 Single-link robot with joint flexibility.



to the first-order solution. 2) The direct solution takes less computation time compared to the indirect approach, which can be attributed to the absence of Lagrange variables. 3) Eliminating the dynamic equations completely from the problem does not necessarily provide the smallest computation time. From Fig. 5, we observe that the indirect solution is more accurate than the direct solution.

B. Flexible Link

A single link manipulator rotating in a vertical plane, driven through a flexible drive train,[7] is shown in Fig. 6. The system has two degrees of freedom, and the equations of motion are

$$I_1 q_1^{(2)} + m_1 g l \sin q_1 + k(q_1 - q_2) = 0 \tag{53}$$

$$I_2 q_2^{(2)} - k(q_1 - q_2) = u \tag{54}$$

where I_1 and I_2 are the moments of inertia of the link and the actuator, m_1 is the mass of the link with its mass center at a distance l from the joint, k is the stiffness of the drive train, g is the gravity constant, and u is the actuator torque. The optimal control problem is to steer the system from a given initial conditions on q_1, q_2, $q_1^{(1)}$, and $q_2^{(1)}$ at t_0 to a specified goal point at t_f while minimizing a cost

$$J = \int_{t_0}^{t_f} u^2 \, \mathrm{d}t$$

Table 2 CPU run-time of example 2 showing direct/indirect and first-order/higher-order comparisons

Systems	CPU run-time, s
Direct first order	10.00
Direct second order	4.700
Direct fourth order	2.380
Direct fourth order with dynamic equations eliminated	8.754
Indirect first order	434.8
Indirect second order	137.8
Indirect fourth order	58.62
Indirect fourth order with dynamic equations eliminated	68.17

The trajectory must satisfy the constraint $-15 \leq u \leq 15$ during motion. The parameters used in the model (MKS units) are $I_1 = I_2 = 1.0$, $k = 1.0$, $g = 9.8$, $m_1 = 0.01$, and $l = 0.5$.

The second-order system, described in Eqs. (53) and (54), can be rewritten in a state-space form as

$$q_1^{(1)} = q_3 \tag{55}$$

$$q_2^{(1)} = q_4 \tag{56}$$

$$q_3^{(1)} = -[m_1 g l \sin q_1 + k(q_1 - q_2)]/I_1 \tag{57}$$

$$q_4^{(1)} = [k(q_1 - q_2) + u]/I_2 \tag{58}$$

where q_1, q_2, q_3, and q_4 are the four state variables.

The two second-order differential equations (53) and (54) have a special structure. From Eq. (53), q_2 can be written explictly in terms of q_1 and its second derivative. On substituting this expression for q_2 into Eq. (54), we obtain a single fourth-order differential equation in q_1 up to its fourth derivative

$$q_1^{(4)} = \alpha_1 u + (\alpha_2 \cos q_1 + \alpha_3) q_1^{(2)} + \alpha_4 \sin q_1 q_1^{(1)^2} + \alpha_5 \sin q_1 \tag{59}$$

where α_i are constants with $\alpha_1 = k/IJ$, $\alpha_2 = -m_1 g l/I$, $\alpha_3 = -k(I + J)/IJ$, $\alpha_4 = m_1 g l/I$, and $\alpha_5 = -m_1 g k l/IJ$.

Additionally, we can define a problem where the expression for the input u from Eq. (59) is explicitly substituted into the cost functional and the constraint. Because the system has several dynamic models, we summarize them briefly as follows: 1) second-order model of Eqs. (53) and (54) with two state variables and one input, 2) first-order model of Eqs. (55–58) with four state variables and one input, 3) fourth-order model of Eq. (59) with one state variable and one input, and 4) fourth-order model with dynamic equations eliminated.

The six-interval direct solution for this example is shown in Fig. 7. The plots for the four cases overlap within the accuracy of Fig. 7. The six-interval indirect solution for this example is shown in Fig. 8 for the four cases, which again overlap within the accuracy of the figure. Because the cost and the constraints do not explicitly contain time, as described in Sec. III, the solution has a first integral. The solution, with the first integral taken explicitly as a constraint, is used to compare the accuracy of solutions obtained from other approaches. Figure 9 shows the plots comparing the accuracy of the solutions in comparison with the solution obtained with the first integral used explicitly as a constraint.

The results for this nonlinear example have similar characteristics to the linear spring–mass–damper system in terms of accuracy and CPU run-time. Table 2 shows the CPU run-time for direct/indirect and first-order/higher-order solution schemes. Some salient points that we observe from Table 2 are as follows: 1) A fourth-order direct or indirect solution has roughly an order of magnitude smaller CPU time compared to the first-order solution. 2) The direct solution is less computation intensive than the indirect approach, and it can be attributed to the absence of Lagrange variables. 3) Eliminating the dynamic equations completely from the problem does not necessarily provide the smallest computation time. From Fig. 9, we observe that the indirect solution is more accurate than the direct solution.

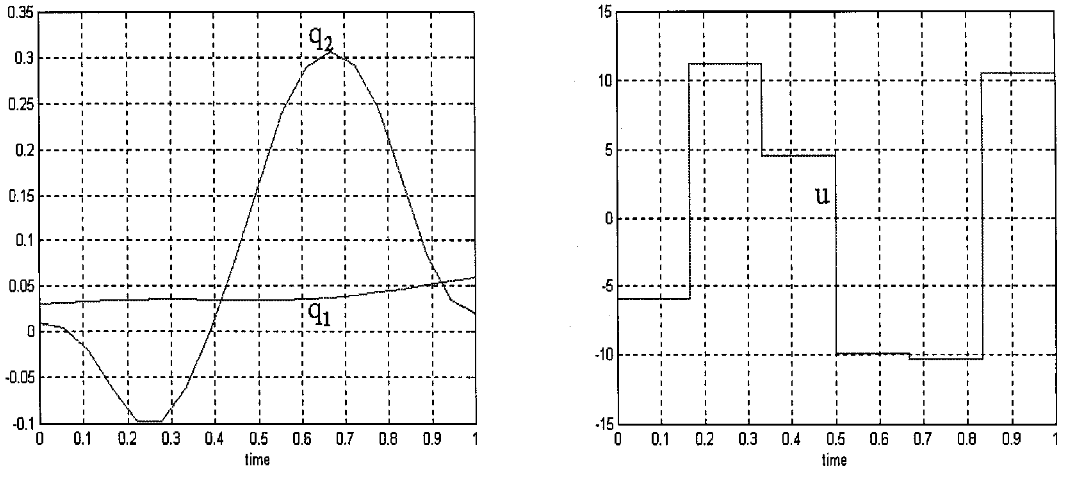


Fig. 7 Six-interval direct solutions for q_1, q_2, and u for example 2; solutions for the cases 1–4 overlap within the accuracy of the figure.

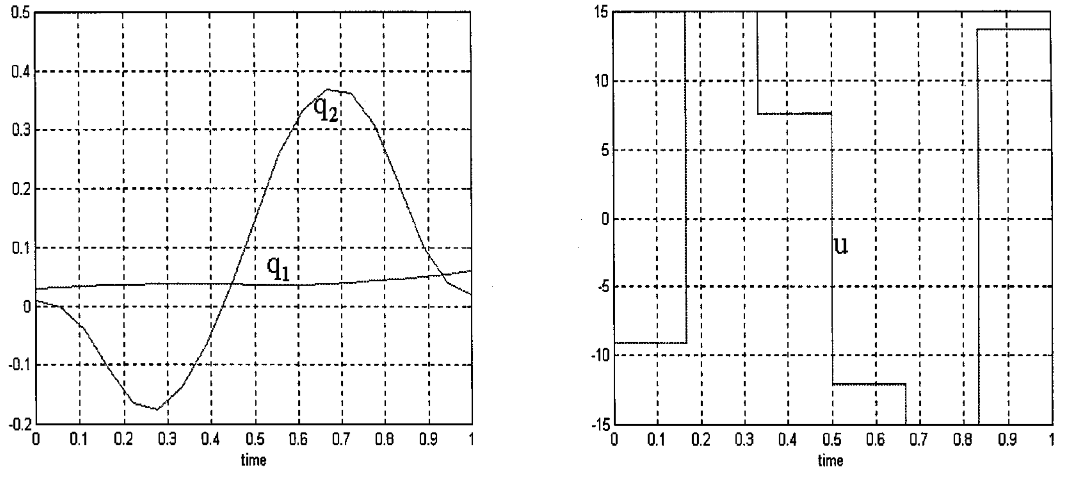


Fig. 8 Six-interval indirect solution for q_1, q_2, and u for example 2; solutions for the cases 1–4 overlap within the accuracy of the figure.

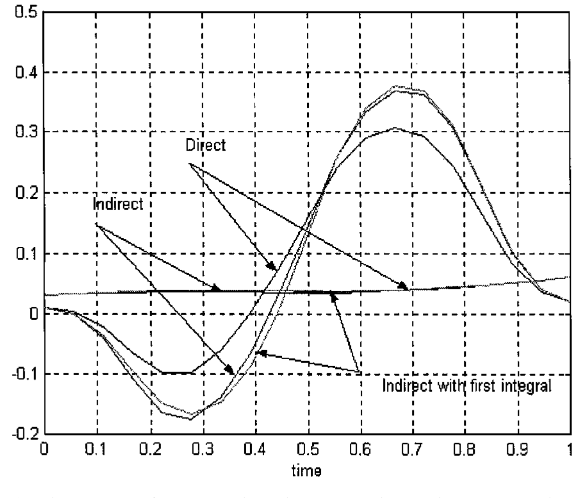


Fig. 9 Accuracy of the solutions in comparison with the solution obtained with the first integral used explicitly as a constraint in example 2.

VI. Conclusions

This paper addressed the underlying theory and computational tools for dynamic optimization of systems in higher-order forms. These higher-order forms either occur naturally in the problem or can be achieved through transformations using linear and nonlinear systems theory. The optimality theory for such higher-order systems was derived using Hamilton–Jacobi equations and calculus of variations. The results from both approaches were compared to yield the same results. The optimality conditions were then used to develop a direct and an indirect computational scheme using MATLAB and a nonlinear programming solver, NPSOL. Through a linear and nonlinear example, it was shown that both with direct and indirect method, an optimization problem posed in the higher-order form requires an order of magnitude less computation compared to a problem posed in the first-order form to yield the same accuracy of the solution. Hence, from a computation efficiency point of view, we believe that a higher-order form of the problem should be pursued.

Acknowledgment

The authors thank National Science Foundation for support of this work through their Presidential Faculty Fellows program.

References

[1]Isidori, A., *Nonlinear Control Systems*, 3rd ed., Springer–Verlag, New York, 1995.

[2]Fliess, M., Levine, J., Martin, P., Rouchon, P., "Flatness and Defect of Nonlinear Systems: Introductory Theory and Examples," *International Journal of Control*, Vol. 61, No. 2, 1995, pp. 1327–1361.

[3]Nieuwstadt, M. J., and Murray, R. M., "Real Time Trajectory Generation for Differentially Flat Systems," *International Journal of Robust and Nonlinear Control*, Vol. 18, No. 11, 1998, pp. 995–1020.

[4]Gelfand, I. S., and Fomin, S. V., *Calculus of Variations*, Prentice–Hall, Englewood Cliffs, NJ, 1963.

[5]Agrawal, S. K., and Veeraklaew, T., "A Higher-Order Method for Dynamic Optimization of a Class of Linear Time-Invariant," *Journal of Dynamic Systems, Measurements, and Control*, Vol. 118, No. 4, 1996, pp. 786–791.

[6]Agrawal, S. K., and Xu, X., "A New Procedure for Optimization of a Class of Linear Time-Varying Systems," *Journal of Vibration and Control*, Vol. 3, No. 4, 1997, pp. 379–396.

[7]Agrawal, S. K., and Faiz, N., "A New Efficient Method for Optimization of a Class of Nonlinear Systems Without Lagrange Multipliers," *Journal of Optimization Theory and Applications*, Vol. 97, No. 1, 1998, pp. 11–28.

[8]Kirk, D. E., *Optimal Control Theory: An Introduction*, Electrical Engineering Series, Prentice–Hall, Englewood Cliffs, NJ, 1970.

[9]Bellman, R. E., and Dreyfus, S. E., *Applied Dynamic Programming*, Princeton Univ. Press, Princeton, NJ, 1962.

[10]Bryson, A. E., Jr., and Ho, Y. C., *Applied Optimal Control*, Hemisphere, New York, 1975.

[11]Berkovitz, L. D., "Variational Methods in Problems of Control and Programming," *Journal of Mathematical Analysis and Applications*, Vol. 3, 1961, pp. 145–169.

[12]Bliss, G. A., *Lectures on the Calculus of Variations*, Univ. of Chicago Press, Chicago, 1946.

[13]Sage, A., *Optimum Systems Control*, Prentice–Hall, Englewood Cliffs, NJ, 1968.

[14]Forsyth, A. R., *Calculus of Variations*, Dover, New York, 1960.

[15]Hahn, D. W., and Johnson, F. T., "Finite Report for Chebyshev Trajectory Optimization Program (CHEBYTOP II)," The Boeing Co., Rept. D180-1296-1, Seattle, WA, June 1971.

[16]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.

[17]Betts, J., Bauer, T., Hoffman, W., and Zondervan, K., "Solving the Optimal Control Problem Using a Nonlinear Programming Technique Part 1: General Formulation," AIAA Paper 84-2037, 1984.