Journal of Industrial and Management Optimization
Vol. 19, No. 12, December 2023, pp. 8602-8616
doi:10.3934/jimo.2023053

A NUMERICAL ALGORITHM FOR CONSTRAINED OPTIMAL CONTROL PROBLEMS

Bowen Zhao[1], Honglei Xu[*1] and Kok Lay Teo[2]

[1]School of Electrical Engineering, Computing and Mathematical Sciences,
Curtin University Perth, 6102, Australia

[2]School of Mathematical Sciences,
Sunway University Kuala Lumpur, 47500, Malaysia

(Communicated by Changjun Yu)

Abstract. In this paper, we consider a general class of discrete-time optimal control problems subject to all-time-step constraints on the state and control variables. The derivations of the gradient formulas for the cost and constraint functions for this constrained discrete-time optimal control problem are rather involved. We present a simple approach to the derivations of these gradient formulas based on reversed automatic differentiation. On this basis, a numerical algorithm is developed to solve this all-time-step constrained discrete-time optimal control problem. We then consider a class of continuous-time optimal control problems subject to continuous state inequality constraints. This constrained continuous-time optimal control problem is discretized into a discrete-time optimal control problem with all-time-step constraints using the Euler discretization method. Then, the algorithm developed for constrained discrete-time optimal control problem is applied to solve this discretized optimal control problem. Numerical examples are presented to verify the applicability of the proposed methods.

1. **Introduction.** Optimal control theory has been applied to solve practical real-world problems in various disciplines, such as engineering [6], environmental sciences (see for example [11] and [5]), and bioprocessing [1]. However, due to the complexity of these practical real-world problems, it is inevitable that they can only be solved by numerical methods. There are various numerical methods that are available in the literature for solving optimal control problems (see for example [9], [3], and [8]). Control parameterization used in conjunction with the constraint transcription method is a popular method for solving constrained optimal control problems.

For discrete-time optimal control problems, they can be solved as nonlinear optimization problems. However, as the governing difference equations are regarded as nonlinear equality constraints, the state and control variables are both considered as decision variables. This will result in having many decision variables and many

2020 *Mathematics Subject Classification.* Primary: 49M52, 49M37; Secondary: 65K05.

Key words and phrases. Discrete-time optimal control problem, continuous-time optimal control problem, automatic differentiation, gradient formulas, numerical algorithm, all-time-step inequality constraints, continuous state inequality constraints.

The second author is supported by the Australian Research Council (LP160100528) and an industry grant (CUB2022).

*Corresponding author: Honglei Xu.

nonlinear equality constraints. In [16], a numerical method is proposed to solve these discrete-time optimal control problems for which only the control variables are regarded as decision variables. For a given control sequence, the corresponding state variables are obtained through solving the difference equations. Then, the gradient formulas for the cost and constraint functions of the problems are derived. Therefore, the problems can be regarded as nonlinear optimization problems and solved using gradient-based optimization methods. The gradient formulas are derived based on the co-state method. Moreover, in [13], two approaches are proposed for the calculation of gradient based on automatic differentiation. In [2], a gradient formula of the cost function is derived based on the reverse automatic differentiation techniques.

In this paper, we consider a general class of discrete-time optimal control problems governed by difference equations and subject to all-time-step inequality constraints. Using the constraint transcription method introduced in the chapter 5 of [12], the all-time-step constraints are approximated by one inequality constraint in the form of the cost function. Then the gradient formulas of the cost and constraint functions can be derived in a unified manner using reversed automatic differentiation [2]. On this basis, a gradient-based numerical algorithm is developed for solving this class of constrained discrete-time optimal control problems.

The rest of the paper is organized as follows. In section 2, the discrete-time optimal control problem subject to all-time-step inequality constraints is formally described. Then, in section 3, through transformation and approximation, the all-time-step inequality constraints are approximated by one terminal state inequality constraint. The cost function is transformed into the cost of terminal state. In section 4, the gradient formulas of the cost and constraint functions are derived in a unified manner using reversed automatic differentiation. On this basis, a numerical algorithm is developed in section 5. The approach is then extended to the continuous-time optimal control problem with continuous state inequality constraints using Euler approximation. Simulation results are presented in section 7. Finally in section 8, some concluding remarks are given.

2. **Problem statement.**

Consider a general class of discrete time dynamical systems described by difference equations in the form of

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(k, \boldsymbol{x}(k), \boldsymbol{u}(k)), \quad k = 0, 1, ..., N-1, \tag{1a}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}^0, \tag{1b}$$

where $\boldsymbol{x} = [x_1, ..., x_n]^T \in \mathbb{R}^n$ and $\boldsymbol{u} = [u_1, ..., u_r]^T \in \mathbb{R}^r$ are, respectively, the state and control vectors, and $\boldsymbol{x}^0 = [x_1^0, ..., x_n^0]^T \in \mathbb{R}^n$ is a given constant vector representing the initial state of the system. Moreover $\boldsymbol{f} = [f_1, \ldots, f_n]^T : \mathbb{Z} \times \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^n$ is a continuously differentiable function with respect to $\boldsymbol{x}$ and $\boldsymbol{u}$, where $\mathbb{Z}$ denotes the set of all non-negative integers. For a control sequence $\boldsymbol{u}(k) = [u_1(k), ..., u_r(k)]^T \in \mathbb{R}^r, k = 0, 1, ..., N-1$, suppose that the following conditions are satisfied:

$$\underline{\boldsymbol{u}} \leq \boldsymbol{u}(k) \leq \overline{\boldsymbol{u}}, \quad k = 0, 1, ..., N-1, \tag{2}$$

where $\underline{\boldsymbol{u}} = [\underline{u}_1, ..., \underline{u}_r]^T \in \mathbb{R}^r$ and $\overline{\boldsymbol{u}} = [\overline{u}_1, ..., \overline{u}_r]^T \in \mathbb{R}^r$ are given constant vectors. Then, this control sequence $\boldsymbol{u}(k) \in \mathbb{R}^r$, $k = 0, 1, ..., N-1$, is referred to as an admissible control sequence. Let $\mathcal{U}$ be the set which contains all such admissible control sequences. A control $\boldsymbol{u} \in \mathcal{U}$ is called a feasible control if it satisfies the following all-time-step inequality constraints:

$$h_i(k, \boldsymbol{x}(k), \boldsymbol{u}(k)) \leq 0, \ \ k = 0, 1, \ldots, N-1; \ i = 1, ..., m, \tag{3}$$

where h_i, $i = 1, ..., m$, are continuously differentiable functions with respect to $\boldsymbol{x}$ and $\boldsymbol{u}$. Let $\mathcal{F}$ denote the set which contains all such feasible controls. We may now state the optimal control problem under consideration in this paper.

Given the dynamic system (1a)-(1b), find a control $\boldsymbol{u} \in \mathcal{F}$ such that the following cost function is minimized over $\mathcal{F}$:

$$g_0(\boldsymbol{u}) = \sum_{k=0}^{N-1} \mathcal{L}_0(k, \boldsymbol{x}(k), \boldsymbol{u}(k)), \tag{4}$$

where $\mathcal{L}_0 : \mathbb{Z} \times \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}$ is a given function which is continuously differentiable with respect to $\boldsymbol{x}$ and $\boldsymbol{u}$. Let this problem be referred to as Problem (P).

3. **Transformation and approximation.** For the cost function (4), define

$$y(k+1) = \sum_{t=0}^{k} \mathcal{L}_0(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \quad k = 0, 1, ..., N-1, \tag{5a}$$

$$y(0) = 0, \tag{5b}$$

which can be written as:

$$y(k+1) = f_0(k, \boldsymbol{x}(k), y(k), \boldsymbol{u}(k)), \quad k = 0, 1, ..., N-1, \tag{6}$$

where

$$f_0(k, \boldsymbol{x}(k), y(k), \boldsymbol{u}(k)) = y(k) + \mathcal{L}_0(k, \boldsymbol{x}(k), \boldsymbol{u}(k)), \tag{6b}$$

Then, (4) is equivalent to

$$y(N), \tag{7}$$

subject to (6).

We now consider the all-time-step inequality constraints (3). Clearly, it is equivalent to

$$\sum_{k=0}^{N-1} \max\{h_j(k, \boldsymbol{x}(k), \boldsymbol{u}(k)), 0\} = 0, \ \ j = 1, ..., m. \tag{8}$$

Since $\max\{\cdot, 0\}$ is non-smooth, we shall apply the constraint transcription technique proposed in [7] and [4] to approximate the non-smooth function $\max\{\cdot, 0\}$ as follows:

$$\gamma_{j,\varepsilon}(k, \boldsymbol{x}(k), \boldsymbol{u}(k)) = \begin{cases} 0, & \text{if } h_j(k, \boldsymbol{x}(k), \boldsymbol{u}(k)) < -\varepsilon \\ \frac{(h_j(k, \boldsymbol{x}(k), \boldsymbol{u}(k)) + \varepsilon)^2}{4\varepsilon}, & \text{if } -\varepsilon \leq h_j(k, \boldsymbol{x}(k), \boldsymbol{u}(k)) \leq \varepsilon \\ h_j(k, \boldsymbol{x}(k), \boldsymbol{u}(k)), & \text{if } h_j(k, \boldsymbol{x}(k), \boldsymbol{u}(k)) \geq \varepsilon \end{cases} \tag{9}$$

where ε is a sufficient small positive number. Define, for each $j = 1, ..., m$,

$$z_j(k+1) = \sum_{t=0}^{k} \gamma_{j,\varepsilon}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \ \ k = 0, 1, ..., N-1, \tag{10a}$$

$$z_j(0) = 0, \tag{10b}$$

which can be written as

$$z_j(k+1) = f_{n+j,\varepsilon}(k, \boldsymbol{x}(k), z_j(k), \boldsymbol{u}(k)), \ \ k = 0, 1, ..., N-1, \tag{11a}$$

where

$$f_{n+j,\varepsilon}(k, \boldsymbol{x}(k), z_j(k), \boldsymbol{u}(k)) = z_j(k) + \gamma_{j,\varepsilon}(k, \boldsymbol{x}(k), \boldsymbol{u}(k)), \ \ j = 1, ..., m. \tag{11b}$$

Now, in view of [4], it follows that the all-time-step inequality constraints (3) will be satisfied if the following inequality constraints are satisfied.

$$z_j(N) - \frac{\varepsilon}{4} \leq 0, \;\; j = 1, ..., m. \tag{12}$$

Now the discrete-time optimal control problem with all-time-step inequality constraints (3) is approximated by the discrete-time optimal control problem subject to terminal state inequality constraints (12). More specifically, the approximate discrete-time optimal control problem may be detailed as follows:

Given the dynamical system

$$\boldsymbol{X}(k+1) = \boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k)), \tag{13a}$$

$$\boldsymbol{X}(0) = \boldsymbol{X}^0, \tag{13b}$$

where $\boldsymbol{X} = [y, x_1, ..., x_n, z_1, ..., z_m]^T$, $\boldsymbol{X}^0 = [0, x_1^0, ..., x_n^0, 0, ..., 0]^T$, and $\boldsymbol{F} = [f_0, f_1, ..., f_n, f_{n+1,\varepsilon}, ..., f_{n+m,\varepsilon}]^T$, find a control $\boldsymbol{u} \in \mathcal{U}$ such that the cost function (7) is minimized subject to the terminal inequality constraints (12). Let this problem be referred to as Problem (Q).

4. **Gradient computation.** Problem (Q) can be solved by using gradient-based optimization techniques, such as the sequential quadratic programming approach (SQP). This family of methods require the gradient information of the cost function (7) and the constraint functions (12). We note that the cost function and the constraint functions are of the same form, and hence the derivations of their gradient formulas can be carried out in a unified manner. The technique used in chapter 7 of [12] is rather involved. A simpler derivation based on reversed automatic differentiation is given below. To derive the gradient formula for the cost function (7), define the Lagrangian given below:

$$\begin{aligned} G_0(k, \boldsymbol{X}, \boldsymbol{u}, \boldsymbol{\Lambda}) = y(N) - \sum_{k=0}^{N-1} [\boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k)) - \\ \boldsymbol{X}(k+1)]^T \boldsymbol{\Lambda}(k+1) + (\boldsymbol{X}^0 - \boldsymbol{X}(0))\boldsymbol{\Lambda}(0), \end{aligned} \tag{14}$$

where

$$\boldsymbol{\Lambda} = [\Lambda_0, \Lambda_1, ..., \Lambda_n, \Lambda_{n+1}, ..., \Lambda_{n+m}]^T \tag{15}$$

is the multiplier function, which is often referred to as the adjoint function.

Theorem 4.1 (Objective gradient)**.** *Consider Problem (Q). The gradient of the cost function* (7) *is*

$$\begin{aligned} \frac{\partial y(N)}{\partial \boldsymbol{u}(k)} &= \frac{\partial G_0(k, \boldsymbol{X}(k), \boldsymbol{u}(k), \boldsymbol{\Lambda}(k+1))}{\partial \boldsymbol{u}(k)} \\ &= \left[\frac{\partial \boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k))}{\partial \boldsymbol{u}(k)}\right]^T \boldsymbol{\Lambda}(k+1), \; k = 0, 1, ..., N-1, \end{aligned} \tag{16}$$

where the system of augmented state variables are obtained by solving the following augmented state system,

$$\begin{aligned} \boldsymbol{X}(k+1) &= \frac{\partial G_0(k, \boldsymbol{X}(k), \boldsymbol{u}(k), \boldsymbol{\Lambda}(k+1))}{\partial \boldsymbol{\Lambda}(k+1)} \\ &= \boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k))), \; k = 0, 1, ..., N-1, \end{aligned} \tag{17a}$$

$$\boldsymbol{X}(0) = \boldsymbol{X}^0, \tag{17b}$$

and the corresponding adjoints are obtained by solving the following adjoint system,

$$\begin{aligned}\boldsymbol{\Lambda}(k) &= \frac{\partial G_0(k, \boldsymbol{X}(k), \boldsymbol{u}(k), \boldsymbol{\Lambda}(k+1))}{\partial \boldsymbol{X}(k)} \\ &= \left[\frac{\partial \boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k))}{\partial \boldsymbol{X}(k)}\right]^T \boldsymbol{\Lambda}(k+1),\ k = 0, 1, ..., N-1, \qquad (18a)\end{aligned}$$

$$\boldsymbol{\Lambda}(N) = \frac{\partial y(N)}{\partial \boldsymbol{X}(N)}. \qquad (18b)$$

Proof. The dynamic system (13) is written in the equality constraints given below:

$$\boldsymbol{\Phi}(k+1) = \boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k)) - \boldsymbol{X}(k+1) = 0,\ \ k = 0, 1, ..., N-1, \qquad (19a)$$

$$\boldsymbol{\Phi}(0) = \boldsymbol{X}^0 - \boldsymbol{X}(0) = 0, \qquad (19b)$$

where $\boldsymbol{\Phi} = [\Phi_y, \Phi_{x_1}, ...\Phi_{x_n}, \Phi_{z_1}, ..., \Phi_{z_m}]^T \in \mathbb{R}^{m+n+1}$.

The Lagrangian for Problem (Q) is given below:

$$\begin{aligned}G_0(k, \boldsymbol{X}, \boldsymbol{u}, \boldsymbol{\Lambda}) = y(N) - \sum_{k=0}^{N-1} &[\boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k)) \\ &- \boldsymbol{X}(k+1)]^T \boldsymbol{\Lambda}(k+1) + (\boldsymbol{X}^0 - \boldsymbol{X}(0))\boldsymbol{\Lambda}(0),\end{aligned} \qquad (20)$$

where $\boldsymbol{\Lambda}(k)$ is the Lagrangian multiplier of the objective function (7) with equality constraints specified by (19). Taking the partial derivative of G_0 with respect to $\boldsymbol{X}(k)$, and then let it be equal to zero, i.e., $\partial G_0/\partial \boldsymbol{X}(k) = 0$, we obtain:

$$\boldsymbol{\Lambda}(k) = \left[\frac{\partial \boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k))}{\partial \boldsymbol{X}(k)}\right]^T \boldsymbol{\Lambda}(k+1). \qquad (21a)$$

Clearly, the last multiplier is given by

$$\boldsymbol{\Lambda}(N) = \frac{\partial y(N)}{\boldsymbol{X}(N)}, \qquad (21b)$$

Since the multiplier is accessible, the desired gradient at each time step can be obtained by letting the partial derivative of G_0 with respect to $\boldsymbol{u}(k)$ to be equal to zero, i.e., $\partial G_0/\partial \boldsymbol{u}(k) = 0$,

$$\frac{\partial G_0}{\partial \boldsymbol{u}(k)} = \frac{\partial y(N)}{\partial \boldsymbol{u}(k)} - \left[\frac{\partial \boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k))}{\partial \boldsymbol{u}(k)}\right]^T \boldsymbol{\Lambda}(k+1) = 0, k = 0, 1, ..., N-1. \qquad (22)$$

Rewritten (22) will give the gradient formula (16) and this completes the proof. □

Remark 4.2 (Terminal condition of $\boldsymbol{\Lambda}(k)$)**.** Note that the augmented state variables contain $y(k)$, $x_i(k)$, and $z_j(k)$, the correspondingly adjoint vector also has three parts, namely $\Lambda_0(k)$, $\Lambda_i(k)$, and $\Lambda_{n+j}(k)$. Their terminal conditions are:

$$\begin{aligned}\Lambda(N) &= [\Lambda_0(N), \Lambda_1(N), ..., \Lambda_n(N), \Lambda_{n+1}(N), ..., \Lambda_{n+m}(N)] \\ &= \left[\frac{\partial y(N)}{\partial y(N)}, \frac{\partial y(N)}{\partial x_1(N)}, ..., \frac{\partial y(N)}{\partial x_n(N)}, \frac{\partial y(N)}{\partial z_1(N)}, ..., \frac{\partial y(N)}{\partial z_m(N)}\right] \qquad (23a) \\ &= \left[1, \underbrace{0, ..., 0}_{n}, \underbrace{0, ..., 0}_{m}\right]. \qquad (23b)\end{aligned}$$

We now consider the *j-th* constraint function (12) of Problem (Q), where $j = 1, ..., m$. Define its Lagrangian given below:

$$G_0(k, \boldsymbol{X}, \boldsymbol{u}, \boldsymbol{\Xi}) = \left(z_j(N) - \frac{\varepsilon}{4}\right) - \sum_{k=0}^{N-1} [\boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k)) \\ - \boldsymbol{X}(k+1)]^T \boldsymbol{\Xi}(k+1) + (\boldsymbol{X}^0 - \boldsymbol{X}(0))\boldsymbol{\Xi}(0), \tag{24}$$

where

$$\boldsymbol{\Xi}^j = [\Xi_0^j, \Xi_1^j, ..., \Xi_n^j, \Xi_{n+1}^j, ..., \Xi_{n+m}^j]^T \tag{25}$$

are the multiplier of the *j-th* constraints.

Theorem 4.3 (The gradient of the *j-th* constraint)**.** *Consider Problem (Q). The gradient of the j-th constraint function (12) is given by*

$$\frac{\partial(z_j(N) - \frac{\varepsilon}{4})}{\partial \boldsymbol{u}(k)} = \frac{\partial G_j(k, \boldsymbol{X}(k+1), \boldsymbol{u}(k+1), \boldsymbol{\Xi}^j(k+1))}{\partial \boldsymbol{u}(k)} \\ = \left[\frac{\partial \boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k))}{\partial \boldsymbol{u}(k)}\right]^T \boldsymbol{\Xi}^j(k+1),\ k = 0, 1, ..., N-1, \tag{26}$$

where the system of the augmented state variables are obtained by solving the following augmented state system forward in time,

$$\boldsymbol{X}(k+1) = \frac{\partial G_0(k, \boldsymbol{X}(k), \boldsymbol{u}(k), \boldsymbol{\Xi}(k+1))}{\partial \boldsymbol{\Xi}(k+1)} \\ = \boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k))),\ k = 0, 1, ..., N-1, \tag{27a}$$

$$\boldsymbol{X}(0) = \boldsymbol{X}^0, \tag{27b}$$

and the corresponding adjoint variables are obtained through solving the following adjoint system backward in time, i.e., from $k = N$ to $k = 0$,

$$\boldsymbol{\Xi}^j(k) = \frac{\partial G_j(k, \boldsymbol{X}(k), \boldsymbol{u}(k), \boldsymbol{\Xi}^j(k+1))}{\partial \boldsymbol{X}(k)} \\ = \left[\frac{\partial \boldsymbol{F}(k, \boldsymbol{X}(k), \boldsymbol{u}(k))}{\partial \boldsymbol{X}(k)}\right]^T \boldsymbol{\Xi}^j(k+1),\ k = 0, 1, ..., N-1, \tag{28a}$$

$$\boldsymbol{\Xi}^j(N) = \frac{\partial z_j(N)}{\partial \boldsymbol{X}(N)}. \tag{28b}$$

The proof of Theorem 2 is similar to the proof of Theorem 1, and hence is omitted.

Remark 4.4 (Terminal condition of $\boldsymbol{\Xi}^j(k)$)**.** For the terminal of adjoint system (28b) the adjoint vector for the *j-th* constraint function also contains three parts, namely, $\Xi_0^j(k)$, $\Xi_i^j(k)$, $\Xi_{n+j}^j(k)$. Their terminal conditions are:

$$\Xi^j(N) = [\Xi_0^j(N), \Xi_1^j(N), ..., \Xi_n^j(N), \Xi_{n+1}^j(N), ...\Xi_{n+j}^j(N), ..., \Xi_{n+m}^j(N)] \\ = \left[\frac{\partial z_j(N)}{\partial y(N)}, \frac{\partial z_j(N)}{\partial x_1(N)}, ..., \frac{\partial z_j(N)}{\partial x_n(N)}, \frac{\partial z_j(N)}{\partial z_1(N)}, ..., \frac{\partial z_j(N)}{\partial z_j(N)}, ..., \frac{\partial z_j(N)}{\partial z_m(N)}\right] \tag{29a}$$

$$= \left[\underbrace{0, ..., 0}_{n+j}, 1, \underbrace{0, ..., 0}_{m-j}\right],\ j = 1, ..., m \tag{29b}$$

5. **Numerical algorithm.** The augmented state variables for the cost function (respectively, the constraint functions) are obtained by solving the augmented state system (17a) (respectively, (27a)) with initial condition (17b) (respectively, (27b)) forward in time. The adjoint variables for the cost function (respectively, the constraint functions) are obtained by solving the adjoint system (18a) (respectively, (28a)) with terminal condition (18b) (respectively, (28b)) backward in time.The numerical partial differentiations of the dynamic functions with respected to state and control can be carried out using automatic differentiation. Finally, the gradients for the cost function and the constraint functions are obtained by using the gradient formulas (16) and (26). With these gradient formulas, the Problem (Q) can be solved using any optimization method, such as SQP. The algorithm is given below.

Algorithm 1 Optimization routine

Step 1. Initialize a smoothing parameter ϵ (where $\epsilon > 0$) and choose an initial control sequence u_0.
Step 2. Solve the augmented system (17) forward in time to obtain the augmented state $\boldsymbol{X}(k), k = 0, ..., N$.
Step 3. Solve the adjoint system (18) backward in time to give the adjoint state $\boldsymbol{\Lambda}(k), k = 0, ..., N$.
Step 4. For $j = 1, ..., m$, solve the adjoint system (28) to have the adjoint state $\boldsymbol{\Xi}(k), k = 0, ..., N$.
Step 5. Use (16) to calculate the gradient of the cost function (7) with respect to $\boldsymbol{u}(k)$.
Step 6. For $j = 1, ..., m$, use (26) to compute the gradient of the j-th constraint function (12) with respect to $\boldsymbol{u}(k)$.
Step 7. Apply symbolic partial differentiation to calculate $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{X}}$ and $\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}$.
Step 8. Implement an SQP solver (such as the MATLAB Optimization Toolbox, *fmincon*) to compute the optimal controls and the states.

6. **Extension to continuous problems.** In this section, we consider a general class of continuous-time optimal control problems governed by the following ordinary differential equation defined on the fixed time interval $(0, T]$,

$$\frac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{30}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{u} \in \mathbb{R}^r$ are, respectively, the state and control vectors, and moreover, $\boldsymbol{x}^0$ is a given constant vector representing the initial state of the system. $\boldsymbol{f} : [0, T] \times \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^n$ is a continuously differentiable function with respect to all its arguments. A control function is said to be an admissible control if it satisfies the box constraints specified by (2). Suppose that an admissible control is such that the following continuous state inequality constraints are satisfied.

$$h_j(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \leq 0, \;\; t \in [0, T]; \; j = 1, ..., m, \tag{31}$$

where $h_j, \; j = 1, ..., m$ are real-valued continuously differentiable functions defined on $[0, T] \times \mathbb{R}^n \times \mathbb{R}^r$. Let $\mathcal{F}$ be the set which consists of all the feasible controls. We now state the constrained continuous-time optimal control problem as given below.

Given the dynamical system (30), find a control $\boldsymbol{u} \in \mathcal{F}$ such that the following cost function

$$g_0(\boldsymbol{u}) = \int_0^T \mathcal{L}_0(t, \boldsymbol{x}(t), \boldsymbol{u}(t))dt, \tag{32}$$

is minimized over $\mathcal{F}$. Let this problem be referred to as Problem (CP).

We now apply the Euler scheme to discrete Problem (CP). Let the time interval $(0, T]$ be partitioned into N equal length subintervals. The time grids are $t_i = (i-1)h,\ i = 1, ..., N+1$ and the step length h is

$$h = \frac{T-0}{N}. \tag{33}$$

The discretized version of system (30) is given by

$$\frac{\boldsymbol{x}(k+1) - \boldsymbol{x}(k)}{h} = \boldsymbol{f}(kh, \boldsymbol{x}(k), \boldsymbol{u}(k)),\ k = 0, ..., N-1. \tag{34}$$

By using the left rule of Riemann summation method, the cost function (32) is approximated by

$$g_0(\boldsymbol{u}) = \sum_{k=0}^{N-1} \mathcal{L}_0(kh, \boldsymbol{x}(k), \boldsymbol{u}(k))h. \tag{35}$$

Furthermore, the continuous state inequality constraints are approximated by the following all-time-step inequality constraints specified at the grid points,

$$h_j(kh, \boldsymbol{x}(k), \boldsymbol{u}(k)) \leq 0,\ \ k = 0, 1, \ldots, N-1;\ j = 1, ..., m. \tag{36}$$

The discretized system (34) can be rearranged as follows:

$$\boldsymbol{x}(k+1) = \boldsymbol{x}(k) + h\boldsymbol{f}(kh, \boldsymbol{x}(k), \boldsymbol{u}(k)),\ k = 0, ..., N-1. \tag{37}$$

To be in the same form as in the system dynamics of discrete-time optimal control problem, we let $\overline{\boldsymbol{f}}(k, \boldsymbol{x}(k), \boldsymbol{u}(k)) = \boldsymbol{x}(k) + h\boldsymbol{f}(kh, \boldsymbol{x}(k), \boldsymbol{u}(k))$. Then, (37) can be rewritten as

$$\boldsymbol{x}(k+1) = \overline{\boldsymbol{f}}(kh, \boldsymbol{x}(k), \boldsymbol{u}(k)),\ k = 0, ..., N-1. \tag{38}$$

So, the discretized version of Problem (CP) may be stated as: Given the system (38), find a control sequence $u(k), k = 0, 1, ..., N$, such that the cost function (35) is minimized subject to the all-time-step inequality constraints (36). Let this problem be referred to as Problem (DP).

Clearly, Problem (DP) is in the same form of Problem (P). Thus, the proposed transformation and approximation as well the the corresponding algorithm can be applied to solve Problem (DP).

7. Simulation results.

7.1. Discrete optimal control problems with simple terminal conditions.

This example is taken from [12]. It describes the vertical ascent of a rocket, where the process is represented by a set of difference equations. x_1, x_2, and x_3 are the gross mass of the rocket, the altitude above the surface and the vertical velocity of the rocket, respectively. u represents the mass flow rate, and in each time step, it will not be allowed to exceed the maximum consumption rate of 0.04, nor be allowed the engine to shut down. The unit of distance is per kilometer and the unit of time is in the second. Rockets gain upward lift by burning their own fuel. Assuming that other than the fuel carried by the rocket, the mass of other parts which weigh 20% of the gross mass does not change in the process. In order for the rocket to be able to climb to the maximum altitude while carrying a certain mass of fuel, the rate of mass flow needs to be precisely controlled. Constant gas nozzle velocity and gravity acceleration are set to be 2 and $0.01km/s^2$, respectively.

$$x_1(k+1) = x_1(k), \tag{39a}$$
$$x_2(k+1) = x_2(k) + x_3(k), \tag{39b}$$
$$x_3(k+1) = x_3(k) + \frac{[Vu(k) - Q(x(k)]}{x_1(k)}. \tag{39c}$$

$$Q(x_2(k), x_3(k)) = 0.05e^{(0.01x_2(k)\cdot x_3(k))^2}, k = 0, 1, ..., N-1. \tag{40}$$

As the rocket rises, the height and velocity of the rocket satisfy the aerodynamic drag function defined by (39). The initial state of the rocket is stationary on the ground and the fuel is full, meaning that $x_1(0) = 1, x_2(0) = x_3(0) = 0$. Fuel will be fully consumed when the rocket reaches to the peak. The objective is to find a control such that the maximum altitude is achieved. The problem may be stated more specifically as follows: Given the dynamic system (39), find a control sequence such that the following objective function:

$$x_2(N), \tag{41}$$

is maximized subject to the following terminal state constraint

$$x_1(N) - 0.2 = 0. \tag{42}$$

The problem is easily solved by the newly developed algorithm. The results obtained are shown in Table 1. The control and state trajectory are displaced in Figure 1. After running our algorithm, the simulation approximate the rocket under the given initial state will reach the peak altitude at $36.4515km$ with a final vertical velocity of $0.112546km/s$. The execute time of the algorithm on this example is 1.192085 seconds. The overall experimental results of the proposed algorithm are slightly better than the result given by [12], namely a higher peak altitude and a slightly reduced calculation time.

N	$x_1(N)/(g_1)$	$x_2(N)/(g_0)$	$x_3(N)$	Execution Time(s)
100	0.2	36.4541	0.1125	1.192085

TABLE 1. Results for Example 7.1

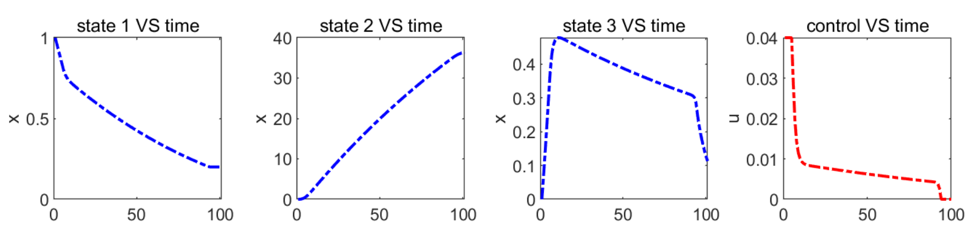

FIGURE 1. State trajectories for Example 7.1

7.2. **Discrete optimal control problem with all-time-step inequality constraint.** The original problem is taken from [10] and it has also been considered in [12]. The dynamic process is described by the following difference equations with initial state $x_1(0) = 0, x_2(0) = -1$:

$$x_1(k+1) = x_1(k) + 0.02x_2(k), \tag{43a}$$

$$x_2(k+1) = 0.98x_2(k) + 0.02u(k), k = 0, 1, ..., N-1. \tag{43b}$$

The objective is to minimize the performance indicator

$$z(k) = \sum_{k=0}^{N-1} \left[(x_1(k))^2 + (x_1(k))^2 + 0.005(u(k))^2 \right], \tag{44}$$

subject to an all-time-step inequality constraint

$$h(k, x_2(k)) = -8(0.02k - 0.05)^2 + x_2(k) + 0.5 \leq 0, k = 0, 1, ..., N-1. \tag{45}$$

After carrying out the transformation and approximation, the approximate version of the above optimal control problem is in the form of Problem (Q). Thus, the algorithm developed in Section 4 is applicable. The results obtained are shown below. Compared with the results given in [10], the algorithm proposed in this paper significantly shortens the calculation time and obtains more optimized calculation results. The calculation time of the proposed algorithm only takes 1/4 of the one shown in the [12] with more accurate results are given.

N	$x_1(N)$	$x_2(N)$	$g_0/y(N)$	$g_1/z_1(N)$	Execution Time(s)
100	-0.22766	0.00824	0.1758267	0	2.208033

TABLE 2. Results for Example 7.2

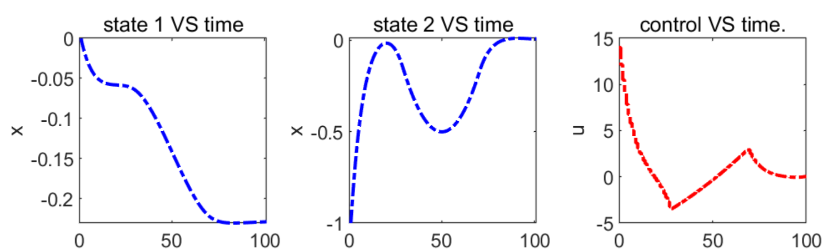

FIGURE 2. State trajectories for Example 7.2

7.3. **Continuous problems.** In this section, we consider a continuous-time optimal control problem taken from [15]. This problem was solved by the adaptive dynamic programming approach proposed in [15]. The same example has also been studied in [14]. The van der Pol oscillator exhibits a limit cycle oscillation phenomenon found in vacuum tube amplifiers, which is a non-conservative oscillator with nonlinear damping. Such system is described as a second-order differential equation with initial state condition of $x(0) = 1.5$ evolving over time,

$$\ddot{x}(t) - \dot{x}(t)(1 - x(t)) + x(t) - u(t).$$

Let x_2 be introduced to describe the velocity. Then the above system is transformed into the system described by the first-order differential equations.

$$\frac{dx_1(t)}{dt} = x_2(t), \tag{46a}$$

$$\frac{dx_2(t)}{dt} = x_2(t)(1 - x_1^2(t)) - x_1(t) + u(t). \tag{46b}$$

Here, x_1 and x_2 are, respectively, displacement and velocity functions of time t. They are referred to as state variables. The control u is a scalar parameter representing the nonlinearity of damping strength. The initial conditions of the state variables are $\boldsymbol{x} = [1.5, 1.5]^T$.

The objective is to find a control such that the following cost function is minimized,

$$J(u) = \frac{1}{2}\int_0^5 \left(x_1^2(t) + x_2^2(t) + u^2(t)\right) dt. \tag{47}$$

For the cost function (47), it can be written in the form given below:

$$J(u) = \frac{1}{2}y(5), \tag{48}$$

where

$$\frac{dy(t)}{dt} = (x_1^2 + x_2^2 + u^2), \tag{49a}$$

$$y(0) = 0. \tag{49b}$$

As the discretization process of continuous-time problem detailed in the Section 6, all the differential equations are now discretized by applying the Euler discretization scheme with the step size $h = (T_{final} - T_{start})/N = (5 - 0)/N$ and the original interval is partitioned into 100 subintervals, i.e., $h = 0.05$ and $N = 100$. The cost function is discretized using left rule of Riemann summation method. Then the transformation and approximation process detailed in Section 3 is carried out to obtain the following discrete-time optimal control problem.

$$\min_u \quad \{J(u) = \frac{1}{2}y(N)\}, \tag{50a}$$

$$\text{s.t.} \quad \begin{aligned} x_1(k+1) &= hx_2(k) + x_1(k), \\ x_2(k+1) &= h[x_2(k)(1 - x_1^2(k)) - x_1(k) + u(k)] + x_2(k), \\ y(k+1) &= y(k) + h[x_1^2(k) + x_2^2(k) + u^2(k)], \\ x_1(0) &= 1.5, x_2(0) = 1.5, y(0) = 0, \end{aligned} \tag{50b}$$

$$k = 0, 1, ..., N-1. \tag{50c}$$

This discrete-time optimal control problem can be solved by the algorithm detailed in Section 4. The results obtained are given below. When the oscillator reaches the terminal point, its displacement will be -0.186942 and it will also have a small velocity of 0.00664405, which is in the same direction of its displacement. The results presented here for the optimal control of the van der Pol oscillator computed by the proposed algorithm are slightly better than the one given in [14], which is 4.338120.

N	$x_1(N)$	$x_2(N)$	$g_0/y(N)$	Execution Time(s)
100	-0.186942	-0.0664405	4.333878	0.52917

TABLE 3. Results for Example 7.3

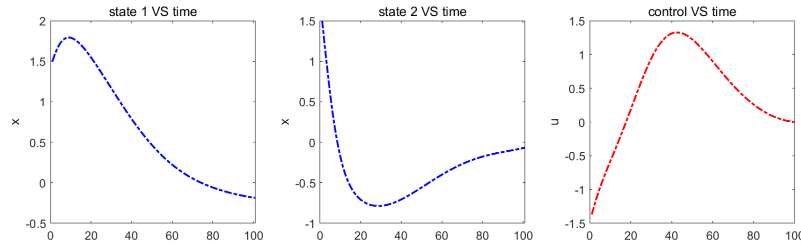

FIGURE 3. State trajectories for Example 7.3

7.4. **Continuous problems with continuous state inequality constraints.** Consider the following problem taken from the Chapter 9 of [12], which is a generalized expression of a continuous mixed state and control inequality constrained Rayleigh problem. For a given system:

$$\frac{dx_1(t)}{dt} = x_2(t),\ x_1(0) = -5, \tag{51a}$$

$$\frac{dx_2(t)}{dt} = -x_1(t) + x_2(t)(1.4 - p(x_2(t))^2) + 4u(t),\ x_2(0) = -5, \tag{51b}$$

in which, $p = 0.14$, find a control sequence u such that the following performance indicator

$$g_0(u) = \int_0^{4.5} ((u(t))^2 + (x_1(t))^2)dt, \tag{52}$$

is minimized subject to a continuous inequality constraint

$$u(t) + \frac{x_1(t)}{6} \leq 0,\ t \in [0, 4.5]. \tag{53}$$

The cost function can be written as

$$J(u) = y(4.5), \tag{54}$$

where

$$\frac{dy(t)}{dt} = u^2 + x_1^2, \tag{55a}$$

$$y(0) = 0. \tag{55b}$$

We apply the discretization procedure described in Section 6 to discrete this problem. All differential equations are discretized into difference equations by applying the Euler discretization scheme with a step size of $h = (T_{final} - T_{start})/N = (4.5-0)/N$. The original interval is divided into 100 subintervals, which is $h = 0.045$ and $N = 100$. After discretizing the cost function using the left rule of Riemann summation method, we convert the continuous inequality constraint on state and control to the inequality constraint for cost and constraint at all the discretized time grid points. The transformation and approximation procedures detailed in Section 3 are then utilized to obtain the following constrained discrete-time optimal control problem,

$$\min_u \quad \{J(u) = y(N)\}, \tag{56a}$$

$$\text{s.t.} \quad \begin{aligned} x_1(k+1) &= h \cdot x_2(k) + x_1(k), \\ x_2(k+1) &= h \cdot [-x_1(k) + x_2(k)(1.4 - p(x_2(k))^2) + 4u(k)] + x_2(k), \\ y(k+1) &= y(k) + h \cdot [(u(k))^2 + (x_1(k))^2], \\ z(k+1) &= z(k) + \gamma_\varepsilon \left(u(k) + \frac{x_1(k)}{6} \right), \end{aligned} \tag{56b}$$

$$\begin{aligned} x_1(0) = -5, x_2(0) = -5, y(0) = 0, \\ k = 0, 1, ...N - 1. \end{aligned} \tag{56c}$$

where

$$\gamma_\varepsilon \left(u(k) + \frac{x_1(k)}{6} \right) = \begin{cases} 0, & \text{if } u(k) + \frac{x_1(k)}{6} < -\varepsilon \\ \frac{\left(u(k) + \frac{x_1(k)}{6} + \varepsilon \right)^2}{4\varepsilon}, & \text{if } -\varepsilon \le u(k) + \frac{x_1(k)}{6} \le \varepsilon \\ \left(u(k) + \frac{x_1(k)}{6} \right), & \text{if } u(k) + \frac{x_1(k)}{6} \ge \varepsilon \end{cases} \tag{57}$$

and ε is a sufficient small positive number.

This transformed constrained discrete-time optimal control problem is now able to be solved by the algorithm detailed in Section 4. The results obtained are given below. The optimal solution attained by the proposed algorithm is 46.01737, and the computational speed is fast even in the presence of a mixed state-control constraint over the entire time horizon. The trajectories presented here also share unique paths with the one in [12]

N	$x_1(N)$	$x_2(N)$	$g_0/y(N)$	Execution Time(s)
100	-0.35361	-1.34045	46.01737	2.36239

TABLE 4. Results for Example 7.4

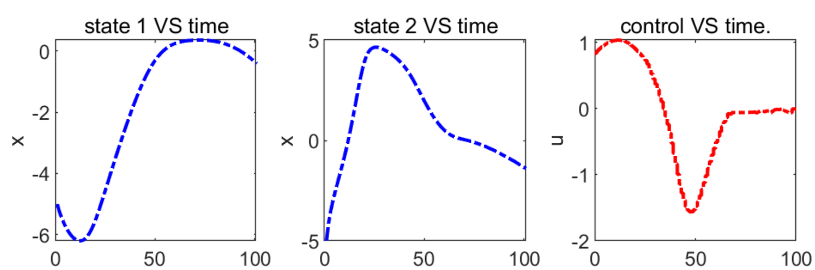

FIGURE 4. State trajectories for Example 7.4

8. **Conclusion.** In this paper, a unified numerical algorithm has been carried out to address the constrained optimal control problem in both discrete and continuous cases. The presented method is developed on the basis of automatic differentiation techniques, which is a systematical application of the chain rule in the derivative calculation process. The proposed algorithm first is used to address the

constrained discrete-time optimal control problem. The discretization of the constrained continuous-time problem and the application of the proposed algorithm on the discretized constrained continuous-time problem have then been introduced.

The novelty of this approach is that it not only reconciles calculation speed with numerical result accuracy for optimal control problems, but it also fully automates the most complicated gradient calculation process of solving the optimal control problem regardless of whether complexity constraints appear or not. Aside from the well-defined problem transformation and approximation techniques in the algorithm, the vast computation requirement for gradient calculation is stored in computer memory in the form of a mathematical expression for cost and constraints function uniformly. It will keep the final result free of errors caused by manually deriving the gradient formula, as was the case in some previous packages. It also enables the iterative calculation procedure to have access to the unique calculation formula with the same precision in every step at the same time for cost and constraints functions.

The proposed gradient formula's theoretical foundation for both the cost and constraints functions is demonstrated. To validate the computational aspect of the proposed algorithm, four different optimal control problems are presented.

REFERENCES

[1] C. Y. Choi and D. D. Perlmutter, Optimal contaminant removal in fermentation and waste treatment processes, *Automatica*, **14** (1978), 325-333.

[2] Y. G. Evtushenko, Fast automatic differentiation and optimal control theory, *Proceedings of the 3rd IFIP WG-7.6 Working Conference on Optimization-Based Computer-Aided Modelling and Design*, Prague, Czech Republic, (1994), 3-12.

[3] C. J. Goh and K. L. Teo, Control parametrization: A unified approach to optimal control problems with general constraints, *Automatica*, **24** (1988), 3-18.

[4] C. J. Goh and K. L. Teo, Alternative algorithms for solving nonlinear function and functional inequalities, *Appl. Math. Comput.*, **41** (1991), 159-177.

[5] E. V. Grigorieva, E. N. Khailov and E. I. Kharitonova, Optimal control of pollution stock through ecological interaction of the manufacturer and the state, *Revista de Matemática: Teoría y Aplicaciones*, **18** (2011), 77-109.

[6] C. Y. F. Ho, B. W. K. Ling, Q. Y. Liu , P. K. S. Tam and K. L. Teo, Optimal PWM control of switched-capacitor DC–DC power converters via model transformation and enhancing control techniques, *IEEE Transactions on Circuits and Systems I: Regular Papers*, **55** (2008), 1382-1391.

[7] L. S. Jennings and K. L. Teo, A computational algorithm for functional inequality constrained optimization problems, *Automatica*, **26** (1990), 371-375.

[8] C. Y. Kaya and J. M. Martínez, Euler discretization and inexact restoration for optimal control, *Journal of Optimization Theory and Applications*, **134** (2007), 191-206.

[9] R. Luus, *Iterative Dynamic Programming*, 2^{nd} edition, Chapman and Hall/CRC, New York, 2000.

[10] R. Mehra and R. Davis, A generalized gradient method for optimal control problems with inequality constraints and singular arcs, *IEEE Transactions on Automatic Control*, **17** (1972), 69-79.

[11] N. H. Shah, S. N. Patel, M. H. Satia and F. A. Thakkar, Optimal control for transmission of water pollutants, *International Journal of Mathematical, Engineering and Management Sciences*, **3** (2018), 381-391.

[12] K. L. Teo, B. Li, C. J. Yu and V. Rehbock, *Applied and Computational Optimal Control*, 1^{st} edition, Springer, New York, 2021.

[13] A. Walther, Automatic differentiation of explicit Runge-Kutta methods for optimal control, *Computational Optimization and Applications*, **36** (2007), 83-108.

[14] X. D. Zhang, S. R. Li and S. L. Lu, Exact derivative calculation for Solving optimal control problem numerically, *Journal of Systems Science and Mathematical Sciences*, **35** (2015), 812-822.

[15] J. Zheng and X. Y. Luo, Adaptive dynamic programming for optimal control of van der pol oscillator, *2018 Chinese Control And Decision Conference (CCDC)*,Shenyang, China, (2018), 1537-1542.

[16] J. Y. Zhou, K. Y. Teo, D. Zhou and G. H. Zhao, Optimal guidance for lunar module soft landing, *Nonlinear Dynamics and Systems Theory*, **10** (2010), 189-201.

Received January 2023; revised March 2023; early access May 2023.