M.M. Gomroki, F. Topputo, F. Bernelli Zazzera, O. Tekinalp
Solving Constrained Optimal Control Problems Using State-Dependent Factorization and Chebyshev Polynomials
Journal of Guidance Control and Dynamics, Vol. 41, N. 3, 2018, p. 618-631
doi:10.2514/1.G002392

Solving Constrained Nonlinear Optimal Control Problems Using State-Dependent Factorization and Chebyshev Polynomials

Mohammad Mehdi Gomroki*

Middle East Technical University, Ankara, 06800, Turkey

University of Turkish Aeronautical Association, Ankara, 06790, Turkey

Francesco Topputo†

Politecnico di Milano, Milano, 20156, Italy

Franco Bernelli-Zazzera‡

Politecnico di Milano, Milano, 20156, Italy

Ozan Tekinalp§

Middle East Technical University, Ankara, 06800, Turkey

The present work introduces a method to solve constrained nonlinear optimal control problems using state dependent coefficient factorization and

*PhD Candidate, Department of Aerospace Engineering, METU. E-mail: mohammad.gomroki@metu.edu.tr and Lecturer, Department of Astronautical Engineering, UTAA. E-mail: mmgomroki@thk.edu.tr

†Assistant Professor, Department of Aerospace Science and Technology, Via La Masa, 34. E-mail: francesco.topputo@polimi.it. AIAA Member.

‡Professor, Department of Aerospace Science and Technology, Via La Masa, 34. E-mail: franco.bernelli@polimi.it. AIAA Senior Member.

§Professor, Department of Aerospace Engineering. AIAA Senior Member. E-mail: tekinalp@metu.edu.tr

Chebyshev polynomials. A recursive approximation technique known as Approximating Sequence of Riccati Equations is used to replace the nonlinear problem by a sequence of linear-quadratic and time-varying approximating problems. The state variables are approximated and expanded in Chebyshev polynomials. Then, the control variables are written as a function of state variables and their derivatives. The constrained nonlinear optimal control problem is then converted to quadratic programming problem, and a constrained optimization problem is solved. Different final state conditions (unspecified, partly specified, and fully specified) are handled, and the effectiveness of the proposed method is demonstrated by solving sample problems.

I. Introduction

Çimen and Banks [1,2] introduced a method known as Approximating Sequence of Riccati Equations (ASRE) which uses State Dependent Coefficient (SDC) factorization and iterative Time-Varying Linear Quadratic Regulator (TV-LQR) approximations to solve Unconstrained Nonlinear Optimal Control (UNOC) problem with unspecified final states. The ASRE approach is applied to many applications like maneuvering of two-craft Coulomb formations at Earth circular orbits and Earth-Moon collinear Libration points [3,4]. Topputo and Bernelli [5,6] solved UNOC problems with unspecified, partly specified, and fully specified final states by using ASRE method differing in the way the time dependent linear quadratic regulator problems are solved. Rather than integrating the Riccati equation in [1,2], the approach represented in [5,6] integrates the Hamiltonian matrix equation to obtain state transition sub-matrices which enables easy handling of boundary conditons.

Many numerical methods have been used to solve nonlinear optimal control problems in the literature. These problems have been solved by using direct and indirect methods [7]. Indirect methods stem from the calculus of variations [8]; direct methods use a nonlinear programming optimization [9]. One of the approaches for handling the direct methods is based on parameterization. For the parameterization method, three different approaches are implemented in the literature: parameterization of the state variables [10], parameterization of the control variables [11], and parameterization of both states and controls [12]. In the current paper, the state variable parameterization approach is implemented to approximate the states with Chebyshev polynomials.

The SDC approaches in [1, 2, 5, 6] involve unconstrained nonlinear optimal control problems. However, Constrained Nonlinear Optimal Control (CNOC) problems are more fit to applications [13–15]. A solution to CNOC problems using Chebyshev polynomials which uses quasilinearization is presented in [13]. Elnagar and Kazemi [14] proposed a method to generate optimal trajectories with linear and nonlinear constrained dynamic systems. Their approach is based on using the Chebyshev polynomials to parameterize the system and transform the optimal control problem to a nonlinear programming problem. Also in [15], a generic Bolza optimal control problem with state and control constraints is solved by using a direct transcription method.

In the present paper, replacing the original dynamic system by TV-LQR problems using iterative ASRE method and parameterizing the states by finite-length Chebyshev polynomials is proposed to convert the constrained nonlinear optimal control problem into a constrained quadratic programming problem. Jaddu and Majdalawi [16] solved nonlinear optimal control problems using SDC factorization and Chebyshev polynomials. Their approach is similar to that in the current paper with three differences. First, there are no constraints on states and controls in their work, the resulting quadratic programming problem has linear equality constraints only, and is more easily solved. Second, further Chebyshev techniques are used in that paper to form an analytical approximation to the performance index, whereas in the current paper numerical integration is used. Third, there are no specified terminal states, while our approach deals with three different final state conditions: hard constrained (final state fully specified), soft constrained (final state not specified), and mixed constrained problems (final state partly specified).

The remainder of the paper is organized as follows. In Section II, the formulation of

UNOC problem is given and two ASRE approaches are recalled that solve this class of unconstrained problems. Although both approaches use SDC factorization form, the first one uses approximating sequence of Riccati equations, and the second one utilizes state transition matrix. In Section III, the proposed method employing SDC factorization and Chebyshev polynomials is introduced; this is called SDC Direct method. The CNOC problem is recalled and reformulated using SDC form. The CNOC problem is converted to quadratic programming problem, and a constrained optimization problem is solved. Section IV presents two case studies that show the effectiveness of the proposed method. Concluding remarks are given in Section V.

II. Review of Optimal Solutions to Unconstrained Nonlinear Optimal Control Problems

In this section, the nonautonomous problems which are nonlinear in the state and linear-affine in the control are considered. The initial state condition is specified, and the final state condition can be either specified or unknown. Both the state variables and controls are unconstrained, and the time span is fixed.

II.A. Statement of Unconstrained Nonlinear Optimal Control Problem

Consider a set of n first-order differential equations

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, t\right) + \mathbf{B}\left(\mathbf{x}, t\right)\mathbf{u} \tag{1}$$

with $\mathbf{f} : \mathbb{R}^{n+1} \to \mathbb{R}^{n}$ and $\mathbf{B} : \mathbb{R}^{n+1} \to \mathbb{R}^{n\times m}$. The goal is to find m control functions $\mathbf{u}(t)$ within the initial and final time, t_0, t_f, such that the performance index

$$J = \varphi\left(\mathbf{x}(t_f), t_f\right) + \int_{t_0}^{t_f} L\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \mathrm{d}t \tag{2}$$

is minimized; $L : \mathbb{R}^{n+m+1} \to \mathbb{R}$, and $\varphi : \mathbb{R}^{n+1} \to \mathbb{R}$. The initial condition is assumed given, namely

$$\mathbf{x}(t_0) = \mathbf{x}_0, \tag{3}$$

while three different forms are examined for the final state condition. These describe the soft constrained problem (SCP), the hard constrained problem (HCP), and the mixed constrained problem (MCP), with the final state not specified, fully specified, and partly specified, respectively. Defined the Hamiltonian $H(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, t) = L(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T [\mathbf{f}(\mathbf{x}, t) + \mathbf{B}(\mathbf{x}, t)\mathbf{u}]$, the solution of the problem may be found using the Euler-Lagrange equations,

$$\dot{\mathbf{x}} = +\frac{\partial H}{\partial \boldsymbol{\lambda}}, \qquad \dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}}, \qquad \frac{\partial H}{\partial \mathbf{u}} = \mathbf{0} \tag{4}$$

in which $\boldsymbol{\lambda}$ is the vector of costates. These Euler-Lagrange equations are the necessary conditions, and their alternative solutions are represented in Eqs. (10)–(14) for the ASRE approach 1 and Eqs. (18)–(19) for the ASRE approach 2.

II.B. Approximating Sequence of Riccati Equations Method

Suppose that $\mathbf{f}(\mathbf{x}, t)$ in Eq. (1) is a continuously differentiable vector-valued function of $\mathbf{x}$ and t in an open set $\boldsymbol{\Gamma} \in \mathbb{R}^{n+1}$, $\mathbf{f}(\cdot) \in \mathcal{C}^1(\boldsymbol{\Gamma})$, and $\mathbf{B}(\mathbf{x}, t) \in \mathcal{C}^0(\boldsymbol{\Gamma})$ is a continuous vector-valued function. In addition, $\mathbf{f}(\mathbf{0}, t) = \mathbf{0}$, $\forall t \in \mathbb{R}$. Under these conditions [17], the State Dependent Coefficient (SDC) factorization of Eq. (1) may be written as

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}, t)\,\mathbf{x} + \mathbf{B}(\mathbf{x}, t)\,\mathbf{u} \tag{5}$$

which is a stabilizable parameterization of the nonlinear system represented in Eq. (1) in a region $\boldsymbol{\Gamma}$ if the pair $\{\mathbf{A}(\mathbf{x}, t),\ \mathbf{B}(\mathbf{x}, t)\}$ is point-wise stabilizable in the linear sense for all

$\mathbf{x} \in \mathbf{\Gamma}$. Redefinition of the objective function (2) in the quadratic-like form is

$$J = \frac{1}{2}\mathbf{x}^T(t_f)S\left(\mathbf{x}(t_f), t_f\right)\ \mathbf{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \left(\mathbf{x}^T Q\left(\mathbf{x}, t\right)\ \mathbf{x} + \mathbf{u}^T R\left(\mathbf{x}, t\right)\ \mathbf{u}\right) \mathrm{d}t \tag{6}$$

where $S\left(\mathbf{x}(t_f), t_f\right)$ and $Q\left(\mathbf{x}, t\right)$ are positive semi-definite, and $R\left(\mathbf{x}, t\right)$ is positive definite time-varying matrices.

II.B.1. Approach 1

The ASRE approach presented in [1, 2] considers the following sequences of Time Varying Linear Quadratic Regulator (TVLQR) approximations

$$\dot{\mathbf{x}}^{[1]} = \mathbf{A}(\mathbf{x}_0)\ \mathbf{x}^{[1]}(t) + \mathbf{B}(\mathbf{x}_0)\ \mathbf{u}^{[1]}(t) \tag{7}$$

$$\dot{\mathbf{x}}^{[k+1]} = \mathbf{A}(\mathbf{x}^{[k]}(t), t)\ \mathbf{x}^{[k+1]} + \mathbf{B}(\mathbf{x}^{[k]}(t), t)\ \mathbf{u}^{[k+1]} \tag{8}$$

where the superscript denotes the iteration. The initial state is $\mathbf{x}^{[k+1]}(t_0) = \mathbf{x}_0$, and the corresponding linear-quadratic cost functional is

$$\begin{aligned} J^{[k+1]} = \frac{1}{2}\left(\mathbf{x}^{[k+1]}(t_f)\right)^T \mathbf{S}(\mathbf{x}^{[k]}(t_f), t_f)\ \left(\mathbf{x}^{[k+1]}(t_f)\right) + \\ \frac{1}{2}\int_{t_0}^{t_f} \left(\mathbf{x}^{[k+1]^T}\mathbf{Q}(\mathbf{x}^{[k]}(t), t)\mathbf{x}^{[k+1]} + \mathbf{u}^{[k+1]^T}\mathbf{R}(\mathbf{x}^{[k]}(t), t)\mathbf{u}^{[k+1]}\right) \mathrm{d}t \end{aligned} \tag{9}$$

Since each approximation is time-varying and linear-quadratic, the optimal control sequence is in the form[1]

$$\mathbf{u}^{[k+1]}(t) = -\mathbf{R}^{-1}(\mathbf{x}^{[k]}(t))\mathbf{B}^T(\mathbf{x}^{[k]}(t))\,\mathbf{P}^{[k+1]}(t)\,\mathbf{x}^{[k+1]}(t) \tag{10}$$

where the real, symmetric and positive-definite matrix $\mathbf{P}^{[k+1]}(t)$ is the solution of

$$\dot{\mathbf{P}}^{[k+1]}(t) = -\mathbf{Q}(\mathbf{x}^{[k]}(t)) - \mathbf{P}^{[k+1]}(t)\,\mathbf{A}(\mathbf{x}^{[k]}(t)) - \mathbf{A}^T(\mathbf{x}^{[k]}(t))\,\mathbf{P}^{[k+1]}(t) + \mathbf{P}^{[k+1]}(t)\,\mathbf{E}(\mathbf{x}^{[k]}(t))\,\mathbf{P}^{[k+1]}(t) \tag{11}$$

with

$$\mathbf{P}^{[k+1]}(t_f) = \mathbf{S}(\mathbf{x}^{[k]}(t_f)) \tag{12}$$

$$\mathbf{E}(\mathbf{x}^{[k]}(t)) = \mathbf{B}(\mathbf{x}^{[k]}(t))\mathbf{R}^{-1}(\mathbf{x}^{[k]}(t))\mathbf{B}^T(\mathbf{x}^{[k]}(t)) \tag{13}$$

Notice that the differential Riccati equation (11) has to be solved backward in time and the optimal state trajectory is obtained by integrating the following differential equation forward in time

$$\dot{\mathbf{x}}^{[k+1]}(t) = \left[\mathbf{A}(\mathbf{x}^{[k]}(t)) - \mathbf{E}(\mathbf{x}^{[k]}(t))\,\mathbf{P}^{[k+1]}(t)\right]\,\mathbf{x}^{[k+1]}(t) \tag{14}$$

II.B.2. Approach 2

The sequence of TVLQR is solved by exploiting the structure of their Euler–Lagrange equations, so avoiding dealing with the matrix differential Riccati equation. This approach is described in [5, 6]. Rather than integrating Eq. (11), the approach in [5, 6] integrates the Hamiltonian matrix equation to obtain state transition sub-matrices that enable easy handling of partially specified terminal states.

Consider the system dynamics and quadratic objective function in Eqs. (5)–(6). The necessary conditions for this problem are obtained by applying Eq. (4), namely

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}, t)\ \mathbf{x} + \mathbf{B}(\mathbf{x}, t)\ \mathbf{u}, \tag{15}$$

$$\dot{\boldsymbol{\lambda}} = -\mathbf{Q}(\mathbf{x}, t)\ \mathbf{x} - \mathbf{A}^T(\mathbf{x}, t)\ \boldsymbol{\lambda}, \tag{16}$$

$$0 = \mathbf{R}(\mathbf{x}, t)\ \mathbf{u} + \mathbf{B}^T(\mathbf{x}, t)\ \boldsymbol{\lambda}. \tag{17}$$

From Eq. (17), one may get

$$\mathbf{u} = -\mathbf{R}^{-1}(\mathbf{x},t)\,\mathbf{B}^T(\mathbf{x},t)\;\boldsymbol{\lambda}, \tag{18}$$

which by substituting into Eqs. (15)–(16) it is possible to get

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\lambda}} \end{pmatrix} = \begin{bmatrix} \mathbf{A}(\mathbf{x},t) & -\mathbf{B}(\mathbf{x},t)\,R^{-1}(\mathbf{x},t)\,\mathbf{B}^T(\mathbf{x},t) \\ -\mathbf{Q}(\mathbf{x},t) & -\mathbf{A}^T(\mathbf{x},t) \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{pmatrix}. \tag{19}$$

The solution of Eq. (19) which is a system of linear differential equations is given by

$$\mathbf{x}(t) = \phi_{xx}(t_0,t)\,\mathbf{x}_0 + \phi_{x\lambda}(t_0,t)\,\boldsymbol{\lambda}_0, \tag{20}$$

$$\boldsymbol{\lambda}(t) = \phi_{\lambda x}(t_0,t)\,\mathbf{x}_0 + \phi_{\lambda\lambda}(t_0,t)\,\boldsymbol{\lambda}_0, \tag{21}$$

where $\mathbf{x}_0$ and $\boldsymbol{\lambda}_0$ are the initial state and costate, respectively. The components of the state transition matrix ϕ_{xx}, $\phi_{x\lambda}$, $\phi_{\lambda x}$, and $\phi_{\lambda\lambda}$ are obtained by integrating the following dynamics

$$\begin{bmatrix} \dot{\phi}_{xx} & \dot{\phi}_{x\lambda} \\ \dot{\phi}_{\lambda x} & \dot{\phi}_{\lambda\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\mathbf{x},t) & -\mathbf{B}(\mathbf{x},t)\,\mathbf{R}^{-1}(\mathbf{x},t)\,\mathbf{B}^T(\mathbf{x},t) \\ -\mathbf{Q}(\mathbf{x},t) & -\mathbf{A}^T(\mathbf{x},t) \end{bmatrix} \begin{bmatrix} \phi_{xx} & \phi_{x\lambda} \\ \phi_{\lambda x} & \phi_{\lambda\lambda} \end{bmatrix}, \tag{22}$$

with the required initial conditions defined as

$$\phi_{xx}(t_0,t_0) = \phi_{\lambda\lambda}(t_0,t_0) = \mathbf{I}_{n\times n}, \quad \phi_{x\lambda}(t_0,t_0) = \phi_{\lambda x}(t_0,t_0) = 0_{n\times n}. \tag{23}$$

The issue here is computing $\boldsymbol{\lambda}_0$ as only $\mathbf{x}_0$ is given. This is given by (refer to [18] for detailed derivation)

$$\begin{aligned}
&\boldsymbol{\lambda}_0(\mathbf{x}_0,\mathbf{x}_f,t_0,t_f) = \phi_{x\lambda}^{-1}(t_0,t_f)\,[\mathbf{x}_f - \phi_{xx}(t_0,t_f)\,\mathbf{x}_0] && \text{for HCP,}\\
&\boldsymbol{\lambda}_0(\mathbf{x}_0,t_0,t_f)=[\phi_{\lambda\lambda}(t_0,t_f)-\mathbf{S}(t_f)\phi_{x\lambda}(t_0,t_f)]^{-1}\,[\mathbf{S}(t_f)\phi_{xx}(t_0,t_f)-\phi_{\lambda x}(t_0,t_f)]\mathbf{x}_0 && \text{for SCP,}\\
&\boldsymbol{\lambda}_0(\mathbf{x}_0,\mathbf{y}_f,t_0,t_f) = (\boldsymbol{\xi}_0(\mathbf{x}_0,\mathbf{y}_f,t_0,t_f),\boldsymbol{\eta}_0(\mathbf{x}_0,\mathbf{y}_f,t_0,t_f)), && \text{for MCP,}
\end{aligned} \tag{24}$$

where $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are the component of $\boldsymbol{\lambda}$ related to the elements of the final state that are partially specified (their expressions are reported in [18]).

III. State-Dependent Coefficient Direct Method

The problems treated in this section are the same as in Section II except that both the states and controls are constrained. The proposed SDC Direct method employs SDC factorization and Chebyshev polynomials. Constrained nonlinear optimal control problem formulation is recalled and reformulated to SDC form. The state variables are approximated and expanded to the Chebyshev polynomials. Then, the state derivatives are derived from the state variables. To this end, the control variables are obtained as a function of state variables and their derivatives. The CNOC problem is converted to quadratic programming problem, and a constrained optimization problem is solved.

III.A. Statement of Constrained Nonlinear Optimal Control Problem

The statement of the constrained problem is similar to that of the unconstrained case in Section II.A, except that this time Eqs. (25)−(26) have to be considered.

$$\mathbf{x}_{\min} \leq \mathbf{x}(t) \leq \mathbf{x}_{\max} \tag{25}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}(t) \leq \mathbf{u}_{\max} \tag{26}$$

The necessary conditions for $\dot{\mathbf{x}}$ and $\dot{\boldsymbol{\lambda}}$ represented in Eq.(4) are kept here, while the third condition, $\partial H/\partial \mathbf{u} = \mathbf{0}$, is replaced by the minimum principle

$$\mathbf{u} = \arg\min_{\mathbf{u}} \mathrm{H}(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, \mathrm{t}) \tag{27}$$

This new form of the necessary condition (minimum principle) on the Hamiltonian prevent the use of alternative solutions described in Eqs. (10)–(14) and Eqs. (18)–(19). Hence, the SDC Direct method to solve the constrained problems in needed.

III.B. Converting Constrained Nonlinear Optimal Control Problems to Quadratic Programming Problems Using Chebyshev Polynomials

From now, and without any loss of generality, we assume $t_0 = 0$. In order to use Chebyshev polynomials, the transformation time $\tau = 2t/t_f - 1$ is used; this is defined in $[-1, 1]$. By using Chebyshev time transformation, TVLQR approximations in Eq. (8) are written as

$$\frac{\mathrm{d}\mathbf{x}^{[k+1]}}{\mathrm{d}\tau}(\tau) \; = \frac{t_f}{2}\left[\mathbf{A}(\mathbf{x}^{[k]}(\tau), \tau)\; \mathbf{x}^{[k+1]} + \mathbf{B}(\mathbf{x}^{[k]}(\tau), \tau)\; \mathbf{u}^{[k+1]}\right] \tag{28}$$

The other equations are written as

$$\begin{aligned} J^{[k+1]} = & \frac{1}{2}\left(\mathbf{x}^{[k+1]}(1)\right)^T \mathbf{S}(\mathbf{x}^{[k]}(1), 1)\; \left(\mathbf{x}^{[k+1]}(1)\right) + \\ & \frac{t_f}{4}\int_{-1}^{1}\left(\mathbf{x}^{[k+1]^T}\mathbf{Q}(\mathbf{x}^{[k]}(\tau), \tau)\; \mathbf{x}^{[k+1]} + \mathbf{u}^{[k+1]^T}\mathbf{R}(\mathbf{x}^{[k]}(\tau), \mathbf{u}^{[k]}(\tau), \tau)\; \mathbf{u}^{[k+1]}\right)\mathrm{d}\tau \end{aligned} \tag{29}$$

$$\mathbf{x}^{[k+1]}(-1) = \mathbf{x}_0 \tag{30}$$

$$\mathbf{x}_{\min} \le \mathbf{x}^{[k+1]}(\tau) \le \mathbf{x}_{\max} \tag{31}$$

$$\mathbf{u}_{\min} \leq \mathbf{u}^{[k+1]}(\tau) \leq \mathbf{u}_{\max} \tag{32}$$

For approximating the state variables, Chebyshev polynomials of first kind, $T_i(\tau)$, are used such that

$$x_j^{[k+1]}(\tau) = {\sum_{i=0}^{N}}' a_i^{(j)} T_i(\tau) \tag{33}$$

where the dash $\left(\sum{}'\right)$ denotes that the first term in the sum is to be halved, $j = 1, 2, ..., n$ is the number of states, N is the degree of the Chebyshev polynomial, and $a_i^{(j)}$ are unknown parameters. Eq. (33) may be rewritten in matrix form as

$$\begin{bmatrix} x_1^{k+1}(\tau) \\ x_2^{k+1}(\tau) \\ . \\ . \\ . \\ x_n^{k+1}(\tau) \end{bmatrix} = \begin{bmatrix} 0.5a_0^{(1)} & a_1^{(1)} & ... & a_N^{(1)} \\ 0.5a_0^{(2)} & a_1^{(2)} & ... & a_N^{(2)} \\ . & . & ... & . \\ . & . & ... & . \\ . & . & ... & . \\ 0.5a_0^{(n)} & a_1^{(n)} & ... & a_N^{(n)} \end{bmatrix} \begin{bmatrix} T_0(\tau) \\ T_1(\tau) \\ . \\ . \\ . \\ T_N(\tau) \end{bmatrix}$$

Using the Kronecker product yields a convenient notation

$$\mathbf{x}^{[k+1]}(\tau) = \left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau)\right) \mathbf{a} \tag{34}$$

where $\mathbf{T}^T(\tau) = [T_0(\tau), T_1(\tau), ..., T_N(\tau)]$ is an $(1 \times (N+1))$ row vector of Chebyshev polynomials, and $\mathbf{a}^T = [a_0^{(1)}/2, a_1^{(1)}, ..., a_N^{(1)}, a_0^{(2)}/2, ..., a_N^{(2)}, ..., a_0^{(n)}/2, ..., a_N^{(n)}]$ is an $(1 \times n(N+1))$ row vector of unknown parameters. Derivative of the state variables is governed by equation

$$\dot{x}_j^{[k+1]}(\tau) = {\sum_{i=0}^{N}}' a_i^{(j)} \dot{T}_i(\tau) \tag{35}$$

which may be written in matrix form as

$$\dot{\mathbf{x}}^{[k+1]}(\tau) = \left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau)\mathbf{D}^T\right)\mathbf{a} \tag{36}$$

where $\dot{\mathbf{T}} = \mathbf{DT}$ is used here, and $\mathbf{D}$ matrix has a dimension of $(N+1) \times (N+1)$ which is defined as following. The derivative of Chebyshev polynomials is defined in [19] as

$$\frac{\mathrm{d}T_N(\tau)}{\mathrm{d}\tau} = 2N \sum_{\substack{i=0 \\ N-i \ odd}}^{N-1}{}' \; T_i(\tau) \tag{37}$$

From Eq. (37) it can be concluded that the derivative of the Chebyshev polynomials of the first kind may be written as

$$\begin{aligned}
\frac{\mathrm{d}T_1(\tau)}{\mathrm{d}\tau} &= T_0(\tau), \\
\frac{\mathrm{d}T_2(\tau)}{\mathrm{d}\tau} &= 4\ T_1(\tau), \\
\frac{\mathrm{d}T_3(\tau)}{\mathrm{d}\tau} &= 3\ T_0(\tau) + 6\ T_2(\tau) \\
\frac{\mathrm{d}T_4(\tau)}{\mathrm{d}\tau} &= 8\ T_1(\tau) + 8\ T_3(\tau) \\
\frac{\mathrm{d}T_5(\tau)}{\mathrm{d}\tau} &= 5\ T_0(\tau) + 10\ T_2(\tau) + 10\ T_4(\tau) \\
\frac{\mathrm{d}T_6(\tau)}{\mathrm{d}\tau} &= 12\ T_1(\tau) + 12\ T_3(\tau) + 12\ T_5(\tau) \\
&\ \ \vdots
\end{aligned}$$

and from the definition of the Chebyshev polynomials of the first kind in [19], we have $\dfrac{\mathrm{d}T_0(\tau)}{\mathrm{d}\tau} = 0$. So, it may be proved that

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 3 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 8 & 0 & 8 & 0 & 0 & 0 & 0 & \dots & 0 \\ 5 & 0 & 10 & 0 & 10 & 0 & 0 & 0 & \dots & 0 \\ 0 & 12 & 0 & 12 & 0 & 12 & 0 & 0 & \dots & 0 \\ 7 & 0 & 14 & 0 & 14 & 0 & 14 & 0 & \dots & 0 \\ 0 & 16 & 0 & 16 & 0 & 16 & 0 & 16 & \dots & 0 \\ . & . & . & . & . & . & . & . & . & . \end{bmatrix}$$

Also, for the technique represented in Section III.C, it is required to get the second derivative of the state variables using the equation

$$\ddot{x}_j^{[k+1]}(\tau) = \sum_{i=0}^{N}{}' a_i^{(j)} \ddot{T}_i(\tau) \tag{38}$$

which in matrix form may be written as

$$\ddot{\mathbf{x}}^{[k+1]}(\tau) = \left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau)\mathbf{D}^T\mathbf{D}^T\right)\mathbf{a} \tag{39}$$

where $\ddot{\mathbf{T}} = \mathbf{DDT}$ is used here. The second derivative of Chebyshev polynomials is defined in [19] as

$$\frac{\mathrm{d}^2 T_N(\tau)}{\mathrm{d}\tau^2} = \sum_{\substack{i=0 \\ N-i\ even}}^{N-2}{}' (N-i)N(N+i)T_i(\tau) \tag{40}$$

From Eq. (40) it can be concluded that the second derivative of the Chebyshev polynomials of the first kind may be written as

$$
\begin{aligned}
\frac{\mathrm{d}^2 T_2(\tau)}{\mathrm{d}\tau^2} &= 4T_0(\tau), \\
\frac{\mathrm{d}^2 T_3(\tau)}{\mathrm{d}\tau^2} &= 24\ T_1(\tau), \\
\frac{\mathrm{d}^2 T_4(\tau)}{\mathrm{d}\tau^2} &= 32\ T_0(\tau) + 48\ T_2(\tau) \\
\frac{\mathrm{d}^2 T_5(\tau)}{\mathrm{d}\tau^2} &= 120\ T_1(\tau) + 80\ T_3(\tau) \\
&\ \ \vdots
\end{aligned}
$$

and from the definition of the Chebyshev polynomials of the first kind in [19], we have $\dfrac{\mathrm{d}^2 T_0(\tau)}{\mathrm{d}\tau^2} = 0$, and $\dfrac{\mathrm{d}^2 T_1(\tau)}{\mathrm{d}\tau^2} = 0$. So,

$$\mathbf{D}^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 24 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 32 & 0 & 48 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 120 & 0 & 80 & 0 & 0 & 0 & 0 & \dots & 0 \\ 108 & 0 & 192 & 0 & 120 & 0 & 0 & 0 & \dots & 0 \\ 0 & 336 & 0 & 280 & 0 & 168 & 0 & 0 & \dots & 0 \\ 256 & 0 & 480 & 0 & 384 & 0 & 224 & 0 & \dots & 0 \\ . & . & . & . & . & . & . & . & . & . \end{bmatrix}$$

Here, it is assumed that the number of states, n, is equal to the number of inputs, m, and the $\mathbf{B}(\mathbf{x}^{[k]})$ matrix in Eq. (28) is square and invertible. Section III.C will handle the case in which the number of states, n, is greater than the number of inputs, m, and $\mathbf{B}(\mathbf{x}^{[k]})$ is not square and invertible. Now rearranging Eq. (28) gives us the required formula for inputs which is in the form

$$\mathbf{u}^{k+1}(\tau) = \mathbf{B}(\mathbf{x}^{[k]}(\tau))^{-1} \left[\frac{2}{t_f} \left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau) \mathbf{D}^T \right) \mathbf{a} - \mathbf{A}(\mathbf{x}^{[k]}(\tau)) \left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau) \right) \mathbf{a} \right] \tag{41}$$

Without loss of generality, we consider that the matrices Q, R, and S are constant for convenience. Taking into account of all the approximations for states, state derivatives, and inputs and substituting them in Eq. (29) may give

$$\hat{J}^{[k+1]} = \frac{1}{2}\mathbf{a}^T\left(\mathbf{I}_n \otimes T(1)\right)\mathbf{S}\left(\mathbf{I}_n \otimes T^T(1)\right)\mathbf{a} + \frac{t_f}{4}\int_{-1}^{1}\left[\mathbf{a}^T\left(\mathbf{I}_n \otimes \mathbf{T}(\tau)\right)\mathbf{Q}\left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau)\right)\mathbf{a}\right]\mathrm{d}\tau +$$
$$\frac{t_f}{4}\int_{-1}^{1}\left[\frac{2}{t_f}\mathbf{a}^T\left(\mathbf{I}_n \otimes \mathbf{D}\mathbf{T}(\tau)\right) - \mathbf{a}^T\left(\mathbf{I}_n \otimes \mathbf{T}(\tau)\right)\mathbf{A}(\mathbf{x}^{[k]}(\tau))^T\right] \times$$
$$\mathbf{F}(\tau)\left[\frac{2}{t_f}\left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau)\mathbf{D}^T\right)\mathbf{a} - \mathbf{A}(\mathbf{x}^{[k]}(\tau))\left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau)\right)\mathbf{a}\right]\mathrm{d}\tau \quad (42)$$

where $\mathbf{F}(\tau) = (\mathbf{B}(\mathbf{x}^{[k]}(\tau))^{-1})^T\,\mathbf{R}\,\mathbf{B}(\mathbf{x}^{[k]}(\tau))^{-1}$, and $\hat{J}^{[k+1]}$ is an approximate value of $J^{[k+1]}$. Multiplication of the elements in Eq.(42) will give the formula for the approximated objective function in the form

$$\hat{J}^{[k+1]} = \frac{1}{2}\mathbf{a}^T\mathbf{h}_0\,\mathbf{a} + \frac{t_f}{4}\int_{-1}^{1}\left[\mathbf{a}^T\mathbf{h}_1\,\mathbf{a} + \frac{4}{{t_f}^2}\mathbf{a}^T\mathbf{h}_2\,\mathbf{a} + \mathbf{a}^T\mathbf{h}_3\,\mathbf{a} - \frac{2}{t_f}\mathbf{a}^T\mathbf{h}_4\,\mathbf{a} - \frac{2}{t_f}\mathbf{a}^T\mathbf{h}_5\,\mathbf{a}\right]\mathrm{d}\tau \quad (43)$$

where

$$\mathbf{h}_0 = \mathbf{S} \otimes T(1)T^T(1)$$
$$\mathbf{h}_1 = \mathbf{Q} \otimes \mathbf{T}(\tau)\mathbf{T}^T(\tau)$$
$$\mathbf{h}_2 = \mathbf{F}(\tau) \otimes \mathbf{D}\mathbf{T}(\tau)\mathbf{T}^T(\tau)\mathbf{D}^T$$
$$\mathbf{h}_3 = (\mathbf{A}(\mathbf{x}^{[k]}(\tau)))^T\,\mathbf{F}(\tau)\,\mathbf{A}(\mathbf{x}^{[k]}(\tau)) \otimes \mathbf{T}(\tau)\mathbf{T}^T(\tau)$$
$$\mathbf{h}_4 = \mathbf{F}(\tau)\,\mathbf{A}(\mathbf{x}^{[k]}(\tau)) \otimes \mathbf{D}\mathbf{T}(\tau)\mathbf{T}^T(\tau)$$
$$\mathbf{h}_5 = (\mathbf{A}(\mathbf{x}^{[k]}(\tau)))^T\,\mathbf{F}(\tau) \otimes \mathbf{T}(\tau)\mathbf{T}^T(\tau)\mathbf{D}^T$$

The current paper computes the objective function numerically using the pointwise evaluations of the states, their derived derivatives, and the derived controls. From Eq. (34) it can be concluded that the initial boundary condition in (30) may be written as

$$\mathbf{x}^{[k+1]}(-1) = \left(\mathbf{I}_n \otimes T^T(-1)\right)\mathbf{a} = \mathbf{x_0} \tag{44}$$

In addition, by substituting the state and control approximations defined in Eqs. (34) and (41) into Eqs. (31) and (32), one may obtain

$$\mathbf{x}_{\min} \leq \left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau)\right)\mathbf{a} \leq \mathbf{x}_{\max} \tag{45}$$

$$\mathbf{u}_{\min} \leq \mathbf{B}(\mathbf{x}^{[k]}(\tau))^{-1}\left[\frac{2}{t_f}\left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau)\mathbf{D}^T\right)\mathbf{a} - \mathbf{A}(\mathbf{x}^{[k]}(\tau))\left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau)\right)\mathbf{a}\right] \leq \mathbf{u}_{\max} \tag{46}$$

Now we are dealing with a special type of mathematical optimization problem known as Quadratic Programming problem. The goal is minimization of a quadratic function, $\hat{J}^{[k+1]}$, of several variables, $\mathbf{a}$, subject to linear constraints on these variables. Inequality constraints result from the state and control constraints (bounds), and equality constraints result from the state boundary conditions. The unknowns are no longer $\mathbf{x}(t)$, $\mathbf{u}(t)$, but rather the coefficients $\mathbf{a}$. The minimization problem is summarized as follows

$$\begin{aligned} \min_{\mathbf{a}} \quad & \hat{J}^{[k+1]} = \frac{1}{2}\mathbf{a}^T\mathcal{H}\,\mathbf{a} \\ \text{s.t.} \quad & \mathcal{A}\,\mathbf{a} - \mathbf{b} < 0 \\ & \mathcal{A}_{\text{eq}}\,\mathbf{a} - \mathbf{b}_{\text{eq}} = 0 \end{aligned} \tag{47}$$

where the quadratic function $\mathcal{H}$ will be described below for the different cases. The matrices and vectors for inequality constraints are defined as

$$\mathcal{A} = \begin{bmatrix} \mathbf{B}(\mathbf{x}^{[k]}(\tau))^{-1} \left[\frac{2}{t_f} \left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau)\mathbf{D}^T \right) - \mathbf{A}(\mathbf{x}^{[k]}(\tau)) \left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau) \right) \right] \\ -\mathbf{B}(\mathbf{x}^{[k]}(\tau))^{-1} \left[\frac{2}{t_f} \left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau)\mathbf{D}^T \right) - \mathbf{A}(\mathbf{x}^{[k]}(\tau)) \left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau) \right) \right] \\ \left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau) \right) \\ -\left(\mathbf{I}_n \otimes \mathbf{T}^T(\tau) \right) \end{bmatrix} \tag{48}$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{u}_{\max} \\ -\mathbf{u}_{\min} \\ \mathbf{x}_{\max} \\ -\mathbf{x}_{\min} \end{bmatrix} \tag{49}$$

and matrices and vectors for equality constraints are defined as

$$\mathcal{A}_{\text{eq}} = \left[\left(\mathbf{I}_n \otimes T^T(-1) \right) \right] \tag{50}$$

$$\mathbf{b}_{\text{eq}} = \left[\mathbf{x}_0 \right] \tag{51}$$

Consider that Eqs. (50) and (51) are valid for soft constrained problems in which the final state conditions are not specified. The matrices and vectors for equality constraints for hard constrained problems and mixed constrained problems are rewritten in Sections III.B.2 and Section III.B.3, respectively. To summarize the proposed method, for the first iteration, the states, the state matrix, and the control matrix are written as $\mathbf{x}^{[k]} = \mathbf{x}_0$, $\mathbf{A}(\mathbf{x}^{[k]}(\tau), \tau) = \mathbf{A}(\mathbf{x}_0)$, and $\mathbf{B}(\mathbf{x}^{[k]}(\tau), \tau) = \mathbf{B}(\mathbf{x}_0)$, respectively. Furthemore, the TVLQR approximations represented in Eq. (28)

$$\frac{d\mathbf{x}^{[k+1]}}{d\tau}(\tau) = \frac{t_f}{2}[\mathbf{A}(\mathbf{x}^{[k]}(\tau),\tau)\ \mathbf{x}^{[k+1]} + \mathbf{B}(\mathbf{x}^{[k]}(\tau),\tau)\ \mathbf{u}^{[k+1]}]$$

followed by the herein equations in Section III.B, help to understand that for the second, third, and the other iterations, the state variables $\mathbf{x}^{[k+1]}$ are approximated by Chebyshev polynomials and the inputs $\mathbf{u}^{[k+1]}$ will be obtained from the states and their derivatives. Note that $\mathbf{A}(\mathbf{x}^{[k]}(\tau),\tau)$ and $\mathbf{B}(\mathbf{x}^{[k]}(\tau),\tau)$ matrices for each iteration are evaluated by the states of the previous iteration. Next, the optimization problem in (47) with its equality and inequality constrained has to be solved to get the parameters in the $\mathbf{a}^T = [a_0^{(1)}/2, a_1^{(1)}, ..., a_N^{(1)}, a_0^{(2)}/2, ..., a_N^{(2)}, ..., a_0^{(n)}/2, ..., a_N^{(n)}]$ matrix. This optimization problem is a type of quadratic programming problem which is the problem of finding a vector $\mathbf{a}$ that minimizes a quadratic function $1/2\ \mathbf{a}^T\mathcal{H}\,\mathbf{a}$, subject to linear constraints. For implementations and numerical examples, the quadprog syntax of Matlab with the interior-point-convex algorithm is used to solve the problems in the current paper. To this end, the states, the state derivatives and the inputs are evaluated from Eqs. (34), (36), and (41), respectively.

III.B.1. Soft Constrained Problem

The quadratic function required for minimization problem (47) is given by

$$\mathcal{H} = \mathbf{h}_0 + \frac{t_f}{2}\int_{-1}^{1}\left[\mathbf{h}_1 + \frac{4}{{t_f}^2}\mathbf{h}_2 + \mathbf{h}_3 - \frac{2}{t_f}\mathbf{h}_4 - \frac{2}{t_f}\mathbf{h}_5\right] d\tau \tag{52}$$

Since the final state is not specified, the equality constraint's matrices and vectors are as stated in Eqs. (50) and (51).

III.B.2. Hard Constrained Problem

The quadratic function required for minimization problem (47) is in the form of

$$\mathcal{H} = \frac{t_f}{2}\int_{-1}^{1}\left[\mathbf{h}_1 + \frac{4}{{t_f}^2}\mathbf{h}_2 + \mathbf{h}_3 - \frac{2}{t_f}\mathbf{h}_4 - \frac{2}{t_f}\mathbf{h}_5\right]\mathrm{d}\tau \tag{53}$$

where final state condition $\mathbf{x}_f$ is specified in this case. For HCP, the matrices and vectors for equality constraints of the optimization problem in (47) are expressed in the form of

$$\mathcal{A}_{\mathrm{eq}} = \begin{bmatrix} \left(\mathbf{I}_n \otimes T^T(-1)\right) \\ \left(\mathbf{I}_n \otimes T^T(1)\right) \end{bmatrix} \tag{54}$$

$$\mathbf{b}_{\mathrm{eq}} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_f \end{bmatrix} \tag{55}$$

III.B.3. Mixed Constrained Problem

The quadratic function required for minimization problem (47) is given by

$$\mathcal{H} = \mathbf{h}_0 + \frac{t_f}{2}\int_{-1}^{1}\left[\mathbf{h}_1 + \frac{4}{{t_f}^2}\mathbf{h}_2 + \mathbf{h}_3 - \frac{2}{t_f}\mathbf{h}_4 - \frac{2}{t_f}\mathbf{h}_5\right]\mathrm{d}\tau \tag{56}$$

The final state is not fully specified in this case. Let the state to be decomposed as $\mathbf{x} = (\mathbf{y}, \mathbf{z})$, where $\mathbf{y} = (x_1, \ldots, x_r)$ are the the r known components at final time, whereas $\mathbf{z} = (x_{r+1}, \ldots, x_n)$ are the remaining $n - r$ free components at final time. So, the equality constraint conditions may be written as

$$\mathcal{A}_{\mathrm{eq}} = \begin{bmatrix} \left(\mathbf{I}_n \otimes T^T(-1)\right) \\ \left(\mathbf{I}_{r\times n} \otimes T^T(1)\right) \end{bmatrix} \tag{57}$$

$$\mathbf{b}_{\text{eq}} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{y}_f \end{bmatrix} \tag{58}$$

Also, it is important to notice that matrix $\mathbf{S}$ in Eq. (43) for $\mathbf{h}_0$ term, should be updated as

$$\mathbf{S} = \begin{bmatrix} \mathbf{0}_{r\times r} & \mathbf{0}_{r\times(n-r)} \\ \mathbf{0}_{(n-r)\times r} & \mathbf{S}_{(n-r)\times(n-r)} \end{bmatrix}$$

where the terms $\mathbf{0}$'s in above matrix are zero matrices.

III.C. Number of Inputs Less than Number of States

Remember the approximated equation for $\mathbf{u}^{[k+1]}$ represented in (41), Section III.B, which shows that the inverse matrix of $\mathbf{B}(\mathbf{x}^{[k]})$ is required. For the case with $n = m$, the matrix $\mathbf{B}(\mathbf{x}^{[k]})$ is square and the necessary condition for that is to be invertible. However, for the case in which, $n > m$, the matrix $\mathbf{B}(\mathbf{x}^{[k]})$ is not square, so not invertible and the following technique briefly explained in [16] may be used. Consider the dynamic system represented in Eq. (5), for the case $n > m$, to be written as

$$\begin{aligned} \dot{x}_1(t) &= A_{11}x_1(t) + A_{12}x_2(t) + \cdots + A_{1n}x_n(t) \\ &\vdots \\ \dot{x}_{q-1}(t) &= A_{(q-1)1}x_1(t) + A_{(q-1)2}x_2(t) + \cdots + A_{(q-1)n}x_n(t) \\ \dot{x}_q(t) &= A_{q1}x_1(t) + A_{q2}x_2(t) + \cdots + A_{qn}x_n(t) + B_{q1}u_1(t) + B_{q2}u_2(t) + \cdots + B_{qm}u_m(t) \\ &\vdots \\ \dot{x}_n(t) &= A_{n1}x_1(t) + A_{n2}x_2(t) + \cdots + A_{nn}x_n(t) + B_{n1}u_1(t) + B_{n2}u_2(t) + \cdots + B_{nm}u_m(t) \end{aligned} \tag{59}$$

where the subscript q is defined as $q = n - m + 1$, and the A and B terms may be state dependent, and these terms are evaluated with the previous iteration's values at the current iteration. Converting the equations to Chebyshev time domain results in

$$
\begin{aligned}
\dot{x}_1(\tau) &= \tfrac{t_f}{2}\left(A_{11}x_1(\tau) + A_{12}x_2(\tau) + \cdots + A_{1n}x_n(\tau)\right) \\
&\vdots \\
\dot{x}_{q-1}(\tau) &= \tfrac{t_f}{2}\left(A_{(q-1)1}x_1(\tau) + A_{(q-1)2}x_2(\tau) + \cdots + A_{(q-1)n}x_n(\tau)\right) \\
\dot{x}_q(\tau) &= \tfrac{t_f}{2}\left(A_{q1}x_1(\tau) + A_{q2}x_2(\tau) + \cdots + A_{qn}x_n(\tau) + B_{q1}u_1(\tau) + B_{q2}u_2(\tau) + \cdots + B_{qm}u_m(\tau)\right) \\
&\vdots \\
\dot{x}_n(\tau) &= \tfrac{t_f}{2}\left(A_{n1}x_1(\tau) + A_{n2}x_2(\tau) + \cdots + A_{nn}x_n(\tau) + B_{n1}u_1(\tau) + B_{n2}u_2(\tau) + \cdots + B_{nm}u_m(\tau)\right)
\end{aligned} \tag{60}
$$

Now, approximate the states $x_1(\tau), x_2(\tau), \ldots, x_{q-1}(\tau)$ by Chebyshev polynomials as represented in Eq. (35), and then obtain the first and second derivatives of them by using Eqs. (36) and (39). Rearrange the first $q-1$ terms in (59) as following to obtain the states $x_q(\tau), x_{q+1}(\tau), \ldots, x_n(\tau)$.

$$
\begin{aligned}
\tfrac{2}{t_f}\dot{x}_1(\tau) - A_{11}x_1(\tau) - \cdots - A_{1(q-1)}x_{q-1}(\tau) &= A_{1q}x_q(\tau) + \cdots + A_{1n}x_n(\tau) \\
&\vdots \\
\tfrac{2}{t_f}\dot{x}_{q-1}(\tau) - A_{(q-1)1}x_1(\tau) - \cdots - A_{(q-1)(q-1)}x_{q-1}(\tau) &= A_{qq}x_q(\tau) + \cdots + A_{qn}x_n(\tau)
\end{aligned} \tag{61}
$$

By taking the derivative from both sides of Eq. (61), the terms $\dot{x}_q(\tau), \dot{x}_{q+1}(\tau), \ldots, \dot{x}_n(\tau)$ will be obtained. So,

$$
\begin{aligned}
\tfrac{2}{t_f}\ddot{x}_1(\tau) - A_{11}\dot{x}_1(\tau) - \cdots - A_{1(q-1)}\dot{x}_{q-1}(\tau) &= A_{1q}\dot{x}_q(\tau) + \cdots + A_{1n}\dot{x}_n(\tau) \\
&\vdots \\
\tfrac{2}{t_f}\ddot{x}_{q-1}(\tau) - A_{(q-1)1}\dot{x}_1(\tau) - \cdots - A_{(q-1)(q-1)}\dot{x}_{q-1}(\tau) &= A_{qq}\dot{x}_q(\tau) + \cdots + A_{qn}\dot{x}_n(\tau)
\end{aligned} \tag{62}
$$

To this end, from last m equations of (60) the required inputs will be found. These inputs $u_1, u_2, \ldots, u_m$ are obtained from

$$\begin{aligned} \tfrac{2}{t_f}\dot{x}_q(\tau) - A_{q1}x_1(\tau) - \cdots - A_{qn}x_n(\tau) &= B_{q1}u_1(\tau) + B_{q2}u_2(\tau) + \cdots + B_{qm}u_m(\tau) \\ &\vdots \\ \tfrac{2}{t_f}\dot{x}_n(\tau) - A_{n1}x_1(\tau) - \cdots - A_{nn}x_n(\tau) &= B_{n1}u_1(\tau) + B_{q2}u_2(\tau) + \cdots + B_{nm}u_m(\tau) \end{aligned} \tag{63}$$

Then, following the same approach represented in Section III.B gives the updated versions of the Eqs. (43), (46), and (48).

IV. Numerical Simulations

Comparison of different optimal control methods for different SDC factorized nonlinear optimal control problems is summarized in Table 1. Two sample problems with nonlinear dynamics are considered to apply and verify the proposed SDC Direct method. In all the cases, the optimization problem represented in (47) is solved in Matlab using quadprog syntax with interior-point-convex algorithm with Intel Core i5 CPU 2.30 GHz.

Table 1. Comparison of different suboptimal methods for different nonlinear optimal control problems.

Method	SCP	HCP	MCP	Constrained
ASRE [1, 2]	Y	N	N	N
ASRE [5, 6]	Y	Y	Y	N
SDC Direct method	Y	Y	Y	Y

In the present implementations, the convergence is reached when

$$\varepsilon = \|\mathbf{x}^{[k+1]} - \mathbf{x}^{[k]}\|_\infty = \max_{t\in[t_0,t_f]} \{|x_j^{[k+1]}(t) - x_j^{[k]}(t)|, j = 1, \ldots, n\} \leq \text{tol} \tag{64}$$

where ε is error and 'tol' is a prescribed tolerance.

IV.A. Problem 1: Van der Pol Oscillator

This problem is taken from [13,16]. Van der Pol oscillator is a second order dynamical system

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -(x_1^2 - 1)x_2 - x_1 + u$$

Initial states are $x_1(0) = 1$, and $x_2(0) = 0$, and the final time is defined as $t_f = 5$. The weighting matrices are $Q = I_2$ and $R = 1$, and the corresponding objective function is

$$J = \frac{1}{2}\int_0^5 \left(x_1^2 + x_2^2 + u^2\right) \mathrm{d}t$$

For SDC factorization the state and input matrices are chosen in the form of

$$A = \begin{bmatrix} 0 & 1 \\ -(1 + x_1 x_2) & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

In this example, the whole procedure of the proposed method will be shown step by step. Since $n > m$, the technique represented in Section III.C has to be considered. In order to use Chebyshev polynomials, the time interval $[0, 5]$ is transformed to $[-1, 1]$ using the transformation time $\tau = 2t/t_f - 1$. The TVLQR approximations are written as

$$\begin{aligned} \dot{x}_1^{[1]}(t) &= x_2^{[1]}(t) \\ \dot{x}_2^{[1]}(t) &= -\left(1 + x_{10}x_{20}\right) x_1^{[1]}(t) + x_2^{[1]}(t) + u^{[1]}(t) \end{aligned} \tag{65}$$

$$\begin{aligned} \dot{x}_1^{[k+1]}(t) &= x_2^{[k+1]}(t) \\ \dot{x}_2^{[k+1]}(t) &= -\left(1 + {x_1}^{[k]}(t){x_2}^{[k]}(t)\right) x_1^{[k+1]}(t) + x_2^{[k+1]}(t) + u^{[k+1]}(t) \end{aligned} \tag{66}$$

Transforming the Eqs. (65) and (66) to Chebyshev time domain, the equations are written as

$$
\begin{aligned}
\dot{x}_1^{[1]}(\tau) &= \frac{t_f}{2}[x_2^{[1]}(\tau)] \\
\dot{x}_2^{[1]}(\tau) &= \frac{t_f}{2}[-(1+x_{10}x_{20})\,x_1^{[1]}(\tau)+x_2^{[1]}(\tau)+u^{[1]}(\tau)]
\end{aligned}
\tag{67}
$$

$$
\begin{aligned}
\dot{x}_1^{[k+1]}(\tau) &= \frac{t_f}{2}[x_2^{[k+1]}(\tau)] \\
\dot{x}_2^{[k+1]}(\tau) &= \frac{t_f}{2}[-\left(1+{x_1}^{[k]}(\tau){x_2}^{[k]}(\tau)\right)x_1^{[k+1]}(\tau)+x_2^{[k+1]}(\tau)+u^{[k+1]}(\tau)]
\end{aligned}
\tag{68}
$$

For the first iteration, the Eq. (67) is used, and the Eq. (68) is implemented for the next iterations. First, the state x_1 is approximated by Chebyshev polynomials. Second, the state x_2 will be obtained from the derivative of x_1. Third, by double differentiation of x_1, the $\dot{x}_2$ will be obtained as shown in the following

$$\ddot{x}_1^{[1]}(\tau) = \frac{t_f}{2}[\dot{x}_2^{[1]}(\tau)] \tag{69}$$

$$\ddot{x}_1^{[k+1]}(\tau) = \frac{t_f}{2}[\dot{x}_2^{[k+1]}(\tau)] \tag{70}$$

To this end, the input u is evaluated from the second equation of Eqs. (67) and (68). Two different subproblems are considered.

IV.A.1. Soft Constrained Problem

This is a SCP in which the final states are not specified, $x_f = free$. Three different cases are solved and discussed. For case 1, both states and control are unconstrained and a solution for this case is available in [16]. Then, the input is constrained in case 2 and states are still unconstrained. Lastly, in case 3 both states and controls are constrained.

Case 1. In this case, it is assumed that there would be no constraints on the states and inputs. Table 2 represents the results for two different degrees of Chebyshev polynomials, $N = 8$ and $N = 12$. The number of iterations, the value of errors, and the objective function values are given. For both Chebyshev degree values, the optimization problem is terminated after 5 iterations and the value of the objective function is in agreement with that given in [16] which is 1.4493959719 for $N = 15$. Looking at the results in Table 2, it seems that for the unconstrained case, increasing the Chebyshev polynomial degree does not improve the objective value. Figure 1 shows the approximate trajectory and control solutions.

Table 2. SDC-Direct method iterations for Problem1-SCP-Case1

N = 8 (CPU time 5.09 s)			N = 12 (CPU time 8.98 s)		
Iteration	Error	J	Iteration	Error	J
1	1.079319e+00	1.685824e+00	1	1.079237e+00	1.685822e+00
2	4.782503e-02	1.427952e+00	2	4.862783e-02	1.427863e+00
3	2.664431e-03	1.435632e+00	3	3.662316e-03	1.435654e+00
4	1.504402e-04	1.435544e+00	4	2.548840e-04	1.435615e+00
5	1.176594e-05	1.435522e+00	5	4.028370e-05	1.435570e+00

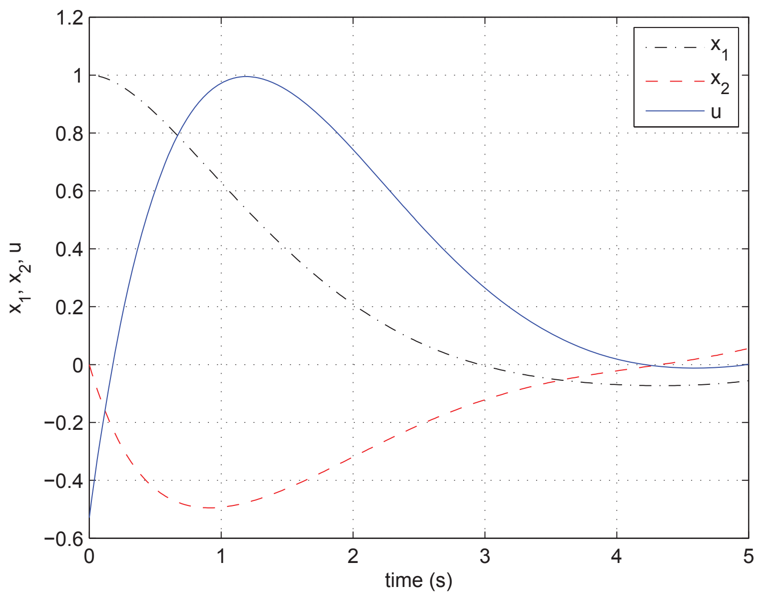


Figure 1. Problem1-SCP-Case1 (N = 12): Approximate trajectory and control solutions.

Case 2. Here we consider that the states are unconstrained and there is constraint on input which is defined as

$$0 \leq u \leq 0.75$$

The iterations, errors, and values of objective functions are given in Table 3. For $N = 8$, the solution is obtained after 10 iterations and for the case when $N = 12$, after 7 iterations the problem is solved. So, it may be concluded that for the case with constraints on inputs, the performance of the algorithm is improved by increasing the degree of Chebyshev polynomials. Moreover, by increasing the Chebyshev polynomial degree, the value of the objective function is decreased. The optimal trajectory and control solutions are shown in Figure 2 for case 1

Table 3. SDC-Direct method iterations for Problem1-SCP-Case2

N = 8 (CPU time 10.86 s)			N = 12 (CPU time 12.7 s)		
Iteration	Error	J	Iteration	Error	J
1	1.668749e+00	3.211873e+00	1	1.451901e+00	3.088773e+00
2	1.444115e+00	1.588960e+00	2	1.269662e+00	1.559349e+00
3	5.368128e-02	1.606897e+00	3	4.048383e-02	1.584761e+00
4	1.837529e-02	1.641353e+00	4	1.046905e-02	1.603692e+00
5	3.591689e-03	1.644906e+00	5	2.010635e-03	1.605551e+00
6	2.828636e-03	1.643633e+00	6	4.903323e-04	1.605002e+00
7	1.530721e-04	1.643527e+00	7	9.863860e-05	1.604865e+00
8	1.570586e-04	1.643605e+00			
9	1.167622e-04	1.643596e+00			
10	9.830951e-06	1.643612e+00			

and case 2 together.

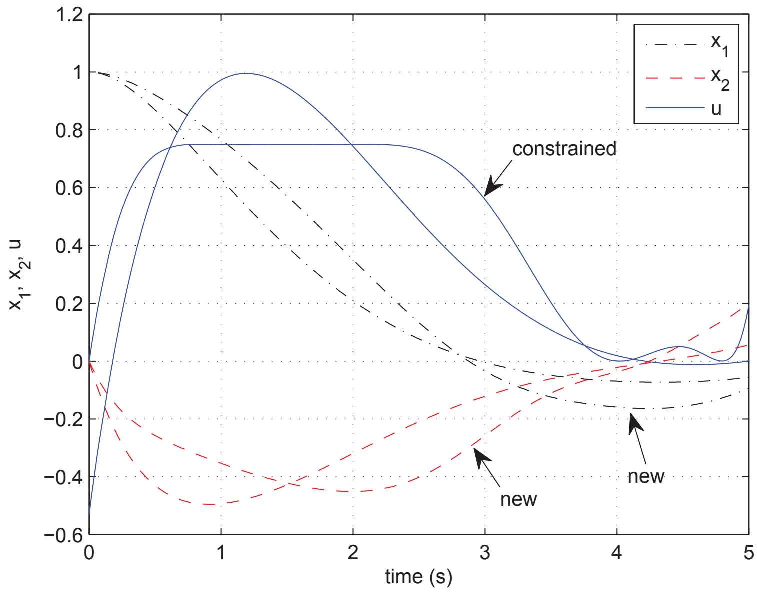


Figure 2. Problem1-SCP-Cases 1 and 2 (N = 12): Approximate trajectory and control solutions.

Case 3. In this case, it is considered to have constraints on both the states and the input. These constraints are defined as

$$0 \leq u \leq 0.75$$

$$0 \leq x_1 \leq 1$$

$$-0.38 \leq x_2$$

The state constraints added to the problem result in more iterations and bigger objective function value for case 3. By considering the results in Table 4, it may be again concluded that the results are improved by increasing the Chebyshev polynomial degree for constrained case. Plots of the cases 1 and 3 are showed in Figure 3. It shows that the constraints on

Table 4. SDC-Direct method iterations for Problem1-SCP-Case3

N = 8 (CPU time 15.03 s)			N = 12 (CPU time 16.81 s)		
Iteration	Error	J	Iteration	Error	J
1	1.762806e+00	5.741236e+00	1	5.878951e+00	9.450530e+01
2	1.303140e+00	2.206247e+00	2	3.879162e+00	9.934352e+00
3	4.327738e-01	1.836886e+00	3	1.987615e+00	1.756250e+00
4	7.568154e-02	1.862552e+00	4	2.819701e-02	1.786414e+00
5	8.307684e-03	1.882555e+00	5	8.817573e-03	1.801913e+00
6	2.916821e-03	1.891366e+00	6	2.809343e-03	1.808637e+00
7	6.889621e-04	1.892950e+00	7	1.023038e-03	1.810733e+00
8	1.223139e-03	1.892241e+00	8	7.394046e-04	1.811066e+00
9	6.099517e-04	1.891546e+00	9	9.837200e-05	1.811003e+00
10	5.788571e-04	1.891260e+00			
11	5.118347e-04	1.891195e+00			
12	1.621073e-04	1.891218e+00			
13	1.334805e-05	1.891245e+00			

the states and control are satisfied and the optimal solutions are changed after applying the constraints.

IV.A.2. Hard Constrained Problem

This is a HCP in which the final states are fully specified; $x_1(5) = -1$, and $x_2(5) = 0$. For this problem, the case with constraints on input is analyzed. Consider the constraint is

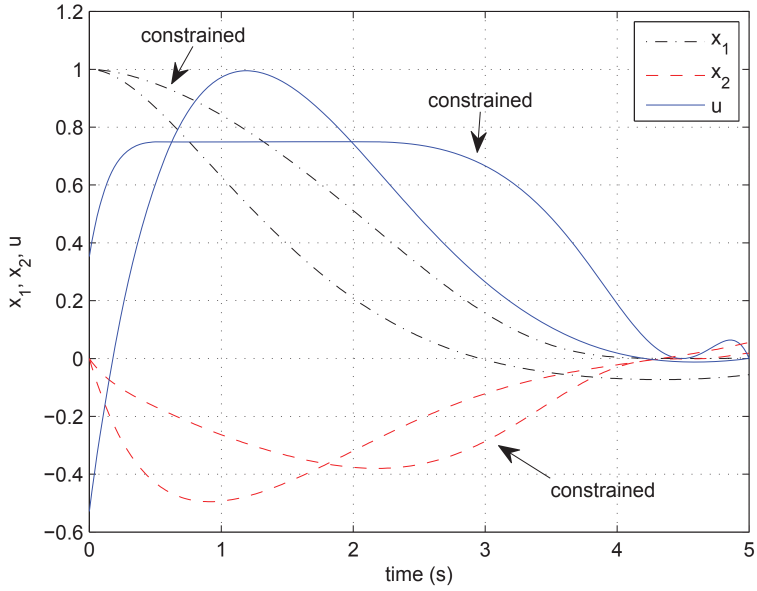


Figure 3. Problem1-SCP-Cases 1 and 3 (N =12): Approximate trajectory and control solutions.

defined as

$$-0.75 \leq u \leq 0.75$$

Figure 4 shows the approximate trajectory and control for unconstrained and constrained cases. It is shown that the initial and final state conditions are satisfied. Moreover, Figure 4 displays that the bounds on control are met, and the new state trajectories are showed after considering the input constraint. A comparison of iteration numbers and the objective function values is represented in Table 5 for different values of Chebyshev polynomial degrees. Again, it may be concluded that the results are improved by increasing the Chebyshev polynomial degree. For the constrained case, a solution exists in [13] and the objective functions of the current paper are in agreement with that given in [13] which is 2.1389 for $N = 12$.

Table 5. SDC-Direct method iterations for Problem1-HCP

N = 8 (CPU time 9.04 s)			N = 12 (CPU time 16.58 s)		
Iteration	Error	J	Iteration	Error	J
1	1.000000e+00	3.130124e+00	1	1.002810e+00	3.249518e+00
2	3.895121e-01	2.144377e+00	2	4.464654e-01	2.116291e+00
3	5.353388e-02	2.146860e+00	3	7.396758e-02	2.126974e+00
4	1.596711e-02	2.173787e+00	4	1.329232e-02	2.147969e+00
5	4.461313e-03	2.179621e+00	5	3.651116e-03	2.151620e+00
6	4.701491e-04	2.179486e+00	6	5.547349e-04	2.151385e+00
7	1.388092e-04	2.179309e+00	7	1.700663e-04	2.151145e+00
8	1.524415e-05	2.179304e+00	8	1.336576e-04	2.151128e+00
			9	7.448995e-06	2.151141e+00

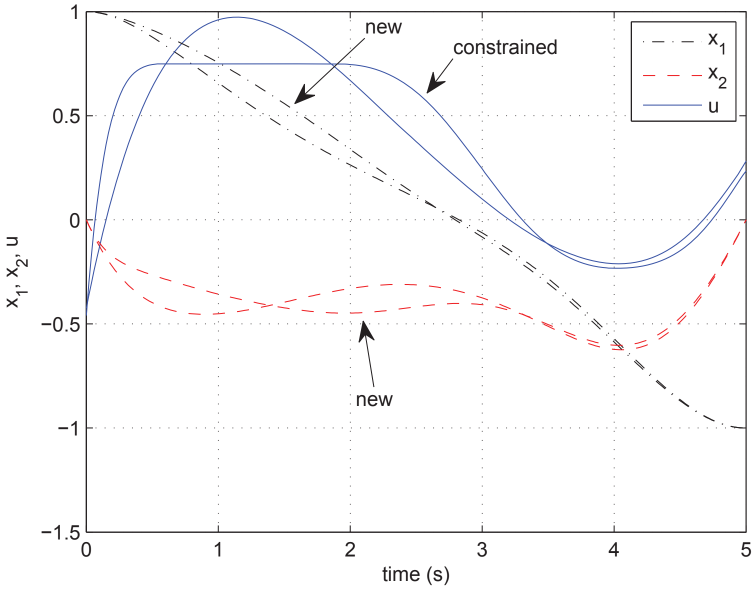


Figure 4. Problem1-HCP (N = 12): Approximate trajectory and control solutions.

IV.B. Problem 2: Low-Thrust Rendezvous

This problem [6,20], considers the planar, relative motion of two particles in a central gravity field expressed in a rotating frame with normalized units: the length unit is equal to the orbital radius, the time unit is such that the orbital period is 2π, and the gravitational parameter is equal to 1. In these dynamics, the state is $\mathbf{x} = (x_1\, x_2\, x_3\, x_4)$; x_1 represents the radial displacement, x_2 represents the tangential displacement, x_3 represents the radial velocity deviations, and x_4 represents the tangential velocity deviations. The control $\mathbf{u} =$

$(u_1\, u_2)$, is made of by the radial and tangential accelerations, respectively. The first order system dynamics are written in the form

$$\begin{aligned}
\dot{x}_1 &= x_3 \\
\dot{x}_2 &= x_4 \\
\dot{x}_3 &= 2x_4 + \left(1 - \frac{1}{r^3}\right)(1 + x_1) + u_1 \\
\dot{x}_4 &= -2x_3 + \left(1 - \frac{1}{r^3}\right) x_2 + u_2
\end{aligned} \tag{71}$$

with $r = \sqrt{(x_1+1)^2 + x_2^2}$. The initial condition is $\mathbf{x}_0 = (0.2, 0.2, 0.1, 0.1)$, and $t_0 = 0$, $t_f = 1$. Since $n > m$, the technique introduced in Section III.C is implemented. For the SDC factorization form, the A and B matrices are chosen as

$$\mathbf{A} = \mathbf{A}(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \left(1 - \frac{1}{r^3}\right)\left(\frac{1}{x_1} + 1\right) & 0 & 0 & 2 \\ 0 & \left(1 - \frac{1}{r^3}\right) & -2 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{72}$$

The objective function is defined as

$$J = \frac{1}{2}\mathbf{x}^T(t_f)\mathbf{S}\,\mathbf{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u}^T\,\mathbf{u}\,\mathrm{d}t. \tag{73}$$

while weighting matrices are $\mathbf{Q} = \mathbf{0}_4$ and $\mathbf{R} = \mathbf{I}_2$. Based on the specification of final states, two different subproblems are considered.

IV.B.1. Soft Constrained Problem

Here the final states are free (SCP), and the weighting matrix for final state conditions is defined as $\mathbf{S} = diag\,(25, 15, 10, 10)$. Three different cases are implemented and the results are given and discussed.

Case 1. This case considers the results of the unconstrained case. Approximate trajectory and control for SDC Direct method are displayed in Figure 5 (dashed lines). The initial state conditions are satisfied and the optimal trajectories and controls are showed. The iterations and objective function values are given in Table 6 for two different degrees of Chebyshev polynomials. The objective function value is the same for both values of N, and these results are in agreement with the solution given in [6] which has the objective function of 0.5660 after 6 iterations.

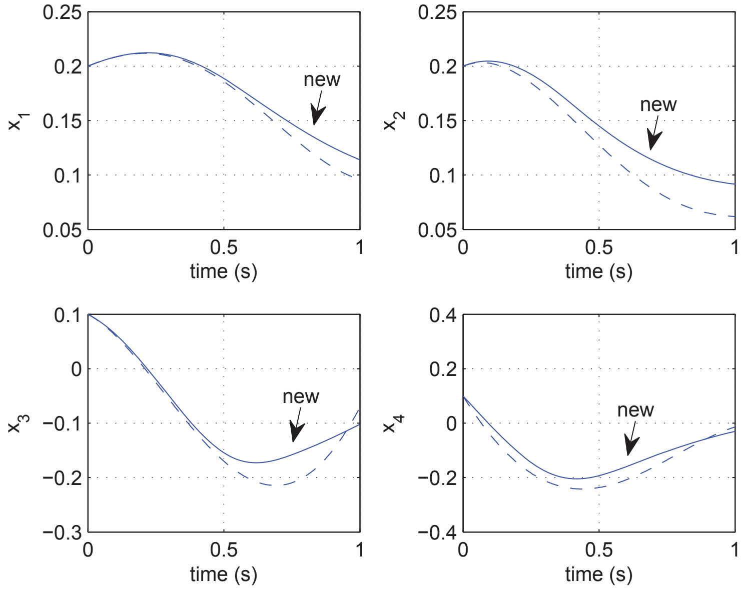
Case 2. Now for the input constrained case, the bounds on the controls are considered as

Table 6. SDC-Direct method iterations for Problem2-SCP-Case1

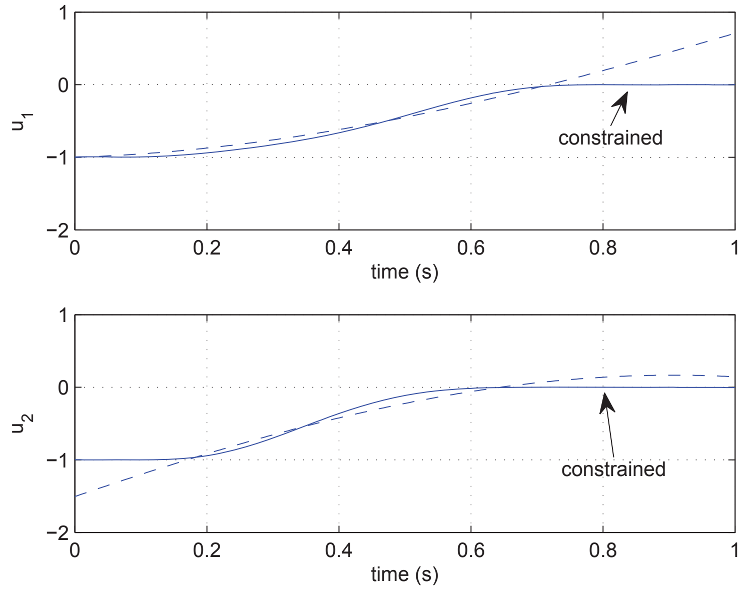
N = 8 (CPU time 8.22 s)			N = 12 (CPU time 14.05 s)		
Iteration	Error	J	Iteration	Error	J
1	3.426168e-01	5.693359e-01	1	3.426165e-01	5.693359e-01
2	1.426767e-03	5.659849e-01	2	1.425440e-03	5.659849e-01
3	1.286568e-05	5.659615e-01	3	1.284486e-05	5.659615e-01

$$-1 \leq u_1 \leq 0, -1 \leq u_2 \leq 0$$

Figure 5 displays the optimal state trajectories and controls with the solid lines. It shows that the input constraints are satisfied and the new plots are the results of these constraints. Table 7 gives the error and objective function values showing an improvement by increasing the Chebyshev polynomial degree.



(a) Approximate trajectory.



(b) Approximate control.

Figure 5. Problem2-SCP-Cases 1 and 2 (N = 12): Approximate trajectory and control solutions.

Case 3. For this case, the states x_3 and x_4 and the controls are constrained as

$$-1 \leq u_1 \leq 0, -1 \leq u_2 \leq 0$$

Table 7. SDC-Direct method iterations for Problem2-SCP-Case2

N = 8 (CPU time 9.23 s)			N = 12 (CPU time 15.09 s)		
Iteration	Error	J	Iteration	Error	J
1	3.003979e-01	6.233571e-01	1	3.057933e-01	6.211742e-01
2	1.911900e-03	6.208723e-01	2	1.876686e-03	6.186383e-01
3	6.134384e-05	6.208419e-01	3	1.782679e-04	6.185371e-01
			4	4.832861e-05	6.185349e-01

$$-0.1 \leq x_3 \leq 0.1$$

$$-0.1 \leq x_4 \leq 0.1$$

Again, for two different Chebyshev degrees, the objective function values are given in Table 8. Also, for this case it may be understood that the performance of the solution can be better by increasing Chebyshev degree. The state and control trajectories are shown in Figure 6 for unconstrained and constrained cases, which demonstrates that the initial state conditions and state constraints are satisfied, and the control trajectories display the justification of the constrained controls.

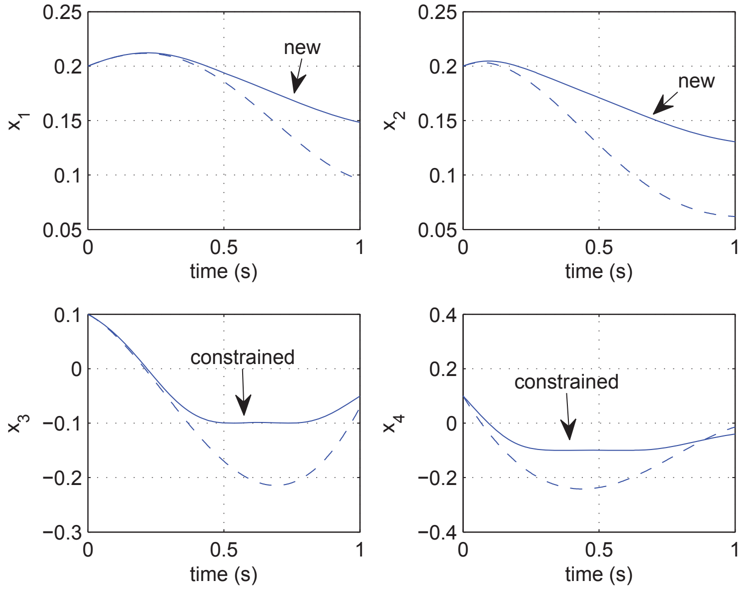
Table 8. SDC-Direct method iterations for Problem2-SCP-Case3

N = 8 (CPU time 9.83 s)			N = 12 (CPU time 16.55 s)		
Iteration	Error	J	Iteration	Error	J
1	1.999999e-01	7.224606e-01	1	2.000000e-01	7.166547e-01
2	1.701514e-03	7.197011e-01	2	1.708269e-03	7.143159e-01
3	1.045409e-04	7.196854e-01	3	3.645138e-04	7.145929e-01
4	1.630952e-06	7.196859e-01	4	1.107111e-05	7.145841e-01

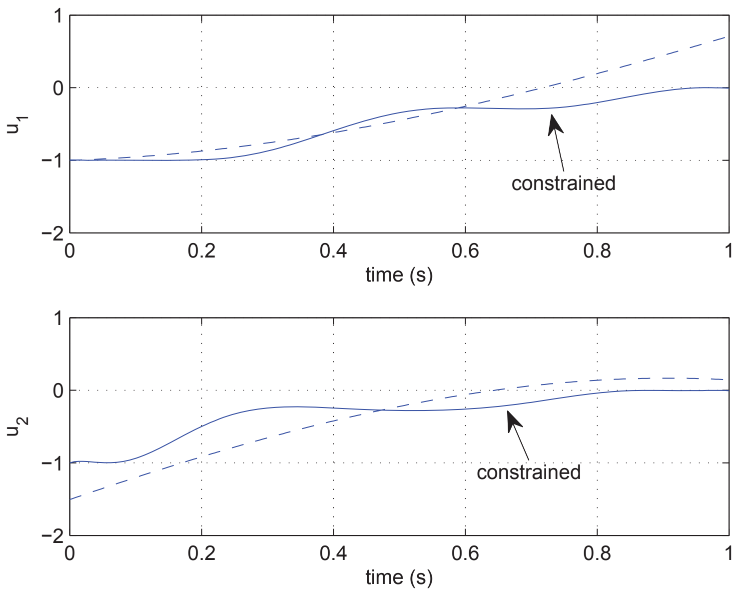
IV.B.2. Hard Constrained Problem

The final states are specified for this type of problem (HCP), and those are given as $\mathbf{x}_f = (0, 0, 0, 0)$. The weighting matrix for final state conditions is considered as $\mathbf{S} = \mathbf{0}$. The unconstrained and constrained results are discussed as following.

Case 1. Unconstrained case is implemented here, and the results are given in Table 9



(a) Approximate trajectory.



(b) Approximate control.

Figure 6. Problem2-SCP-Cases 1 and 3 (N = 12): Approximate trajectory and control solutions with SDC Direct method.

and are showed with dashed lines in Figure 7. The objective function value is in agreement with that given in [6] which is 0.9586 for 5 iterations.

Table 9. SDC-Direct method iterations for Problem2-HCP-Case1

N = 8 (CPU time 8.72 s)			N = 12 (CPU time 13.35 s)		
Iteration	Error	J	Iteration	Error	J
1	4.731670e-01	9.629775e-01	1	4.731672e-01	9.629775e-01
2	9.376290e-04	9.584905e-01	2	9.395097e-04	9.584905e-01
3	2.924892e-06	9.584936e-01	3	2.971336e-06	9.584936e-01

Case 2. For this case, just the inputs are constrained as

$$-1 \leq u_1 \leq 2, -2 \leq u_2 \leq 0$$

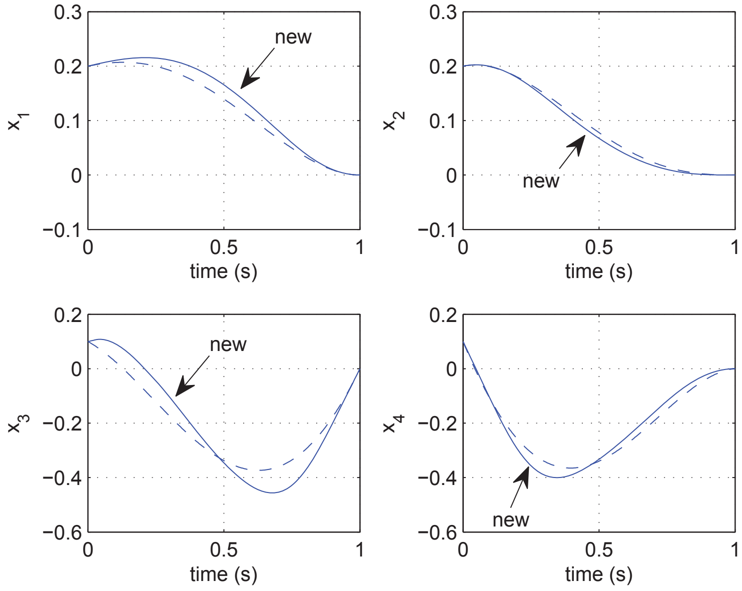
Table 10 represents the number of iterations, errors, and objective function values, and the optimal trajectories are displayed in Figure 7. Notice that for $N = 8$ no optimal solution satisfying the constraints was found.

Table 10. SDC-Direct method iterations for Problem2-HCP-Case2

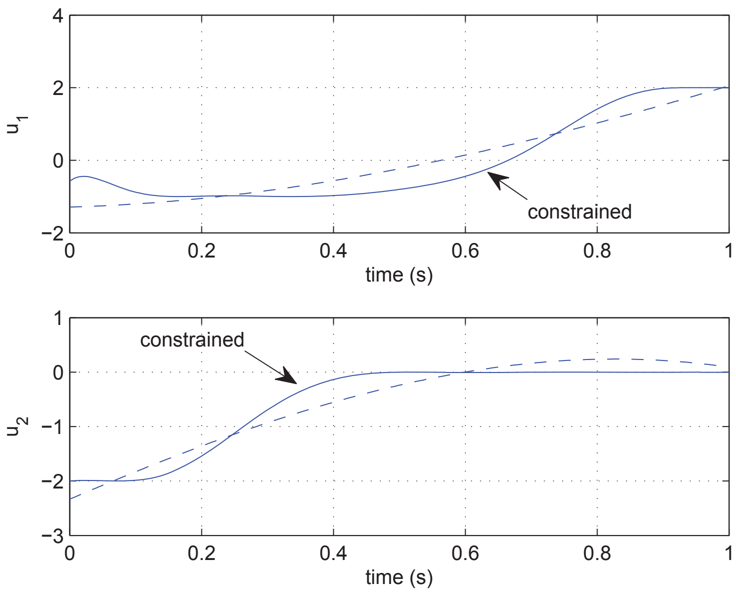
N = 12 (CPU time 37.68 s)		
Iteration	Error	J
1	5.546035e-01	1.072986e+00
2	3.402995e-03	1.069829e+00
3	2.085145e-04	1.069635e+00
4	7.464199e-05	1.069662e+00

V. Conclusions

In the present paper, constrained nonlinear optimal control problems are handled by the proposed approach named as State Dependent Coefficient Direct (SDC-Direct) method which uses SDC factorization and Chebyshev polynomials. It has been shown that two different problems with different final state conditions are solved by the proposed technique. The effectiveness of the method is demonstrated through numerical implementations and using different values for Chebyshev polynomials. Its ability to solve constrained nonlinear optimal control problems is shown.



(a) Approximate trajectory.



(b) Approximate control.

Figure 7. Problem2-HCP-Cases 1 and 2 (N = 12): Approximate trajectory and control solutions with SDC Direct method.

Acknowledgement

This research was supported by the European Union FP7 Marie Curie project AstroNet-II, project number PITN-GA-2011-289240.

References

[1]Çimen, T. and Banks, S., "Global Optimal Feedback Control for General Non Linear System with Non-Quadratic Performance Criteria," *System & Control Letters*, Vol. 53, 2004, pp. 327–346.

[2]Çimen, T. and Banks, S., "Nonlinear Optimal Tracking Control with Application to Super-Tankers for Autopilot Design," *Automatica*, Vol. 40, 2004, pp. 1845–1863.

[3]Gomroki, M. M. and Tekinalp, O., "Maneuvering of Two-Craft Coulomb Formation Using ASRE Method," *Paper AIAA 2014-4164, AIAA/AAS Astrodynamics Specialist Conference, San Diego, California, USA, August 4-7*, 2014.

[4]Gomroki, M. M. and Tekinalp, O., "Nonlinear Control to Maneuver a Two-Craft Coulomb Formation at Libration Points," *Paper AIAA 2015-1555, AIAA Guidance, Navigation, and Control Conference, Kissimmee, Florida, USA, 5–9 Jan.*, 2015.

[5]Topputo, F. and Bernelli-Zazzera, F., "A Method to Solve Nonlinear Optimal Control Problems in Astrodynamics," *Advances in the Astronautical Sciences*, Vol. 145, 2012, pp. 1531–1544.

[6]Topputo, F. and Bernelli-Zazzera, F., "Approximate Solutions to Nonlinear Optimal Control Problems in Astrodynamic," *ISRN Aerospace Engineering*, Vol. 2013, No. Article ID 950912, 2013, pp. 1–7.

[7]Betts, J., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control and Dynamics*, Vol. 21, 1998, pp. 193–207.

[8]Bryson, A. and Ho, Y., *Applied Optimal Control*, John Wiley & Sons, New York, 1975, Chaps. 2 and 5, pp. 65–69, 150–151.

[9]Conway, B. A., "Spacecraft Trajecory Optimization Using Direct Transcription and Nonlinear Programming," *Spacecraft Trajectory Optimization*, edited by B. Conway, Cambridge University Press, 2010, Chap. 3, pp. 37–78, pp. 37–78.

[10]Yen, V. and Nagurka, M., "Optimal control of linearly constrained linear systems via state parametrization," *Optimal Control Applications and Methods*, Vol. 13, 1992, pp. 155–167.

[11]Teo, K. and Wong, K., "Nonlinearly Constrained Optimal Control Problems," *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics*, Vol. 33, 1992, pp. 517–530.

[12]Vlassenbroeck, J. and Van Dooren, R., "A Chebyshev Technique for Solving Nonlinear Optimal Control Problems," *IEEE transactions on automatic control*, Vol. 33, 1988, pp. 333–340.

[13]Jaddu, H., "Direct solution of nonlinear optimal control problems using quasilinearization and Chebyshev polynomials," *Journal of the Franklin Institute*, Vol. 339, 2002, pp. 479–498.

[14]Elnagar, G. N. and Kazemi, M. A., "Pseudospectral Chebyshev optimal control of constrained nonlinear dynamical systems," *Computational Optimization and Applications*, Vol. 11, 1998, pp. 195–217.

[15]Fahroo, F. and Ross, M., "Direct trajectory optimization by a Chebyshev pseudospectral method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, 2002, pp. 160–166.

[16]Jaddu, H. and Majdalawi, A., "An iterative technique for solving a class of nonlinear quadratic optimal control problems using Chebyshev polynomials," *International Journal of Intelligent Systems and Applications*, Vol. 6, 2014, pp. 53–57.

[17]Çimen, T., "Survey of State-Dependent Riccati Equation in Nonlinear Optimal Feedback Control Synthesis," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 4, 2012, pp. 1025–1047.

[18]Topputo, F., Miani, M., and Bernelli-Zazzera, F., "Optimal Selection of the Coefficient Matrix in State-Dependent Control Methods," *Journal of Guidance, Control, and Dynamics*, Vol. 38, 2015, pp. 861–873.

[19]Mason, J. C. and Handscomb, D. C., *Chebyshev polynomials*, CRC Press, 2002.

[20]Park, C., Guibout, V., and Scheeres, D., "Solving Optimal Continuous Thrust Rendezvous Problems with Generating Functions," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 2, March–April 2006, pp. 321–331.