MATHEMATICAL COMMUNICATIONS
Math. Commun. **28**(2023), 153–169

An iterative Legendre pseudospectral method for suboptimal control of nonlinear control affine systems

Mohammad Shirazian*

Department of Mathematics, University of Neyshabur, Khorasan Razavi, 9 319 774 446, Iran

Received April 13, 2022; accepted April 6, 2023

Abstract. The objective of this paper is to present a novel method to design a suboptimal controller for nonlinear control affine systems. The proposed method is a combination of a successive approximation method (SAM) and a Legendre pseudospectral method, used for solving the extreme conditions derived from Pontryagin's maximum principle (PMP). The convergence theorem and the superlinear rate of convergence are presented in this method. Some numerical examples are included to demonstrate the accuracy, efficiency, and reliability of the proposed method.
AMS subject classifications: 49M05, 65K10
Keywords: optimal control problem, successive approximation method, Legendre pseudospectral method, Pontryagin's maximum principle, rate of convergence

1. Introduction

The theory of optimal control encompasses a wide range of applications in a variety of areas, including biology, engineering, social sciences, agriculture, and so on. Lenhart et al. examined optimal control of biological systems, such as infectious diseases, cancer, prey-predator models, etc. [9]. The applications of optimal control in aircraft systems [6], robotics, etc. have also been thoroughly investigated by researchers. The spread of the rumor on social networks has also been studied recently [20, 21, 10]. Modeling of virus propagation among plants and its optimal control by organic pesticides has been investigated in [3].

Methods of solving optimal control problems are generally divided into two categories: direct and indirect methods. Direct methods apply parameterizing or discretizing the state and control variables, which turn the problem into a nonlinear programming problem. The most popular methodologies are pseudospectral methods, where the states and controls are parametrized by some basis polynomials, such as Legendre, Chebyshev, and Jacobi polynomials, and then collocated at some particular points, known as collocation nodes [17]. For instance, the Legendre pseudospectral method is used to derive an optimal strategy for active magnetic bearing systems [14], or it is used to control the spread of tuberculosis, by vaccination and treatment, along with the Chebyshev pseudospectral method [15]. Some other direct methods are the hybrid parametrization approach [1], the radial basis function (RBF) collocation method [12], etc.

*Corresponding author. *Email address:* shirazian@neyshabur.ac.ir (M. Shirazian)

On the other hand, in indirect methods, which are the main subject of this study, we must first write the optimality conditions and then solve them to obtain the optimal states and controls. The optimality conditions come from Pontryagins maximum principle or dynamic programming. In this paper, we consider the following affine in control nonlinear dynamical system:

$$\begin{aligned} &\dot{x}(t) = f(t, x(t)) + g(t, x(t))u(t), \quad t \in [0, t_F] \\ &x(0) = x^0, \end{aligned} \tag{1}$$

where $x(t)$ and $u(t)$ are the state and control variables which have n and m components, respectively. In addition, $f(t, x(t))$ and $g(t, x(t))$ are two continuously differentiable functions in all their arguments. Our goal is to minimize the objective functional:

$$J[x, u] = \frac{1}{2} \int_0^{t_F} (Q(x(t)) + u^T(t) R u(t)) dt,$$

subject to dynamical system (1), for $Q(x(t))$ a positive semi-definite real function and $R \in \mathbb{R}^{m \times m}$ a positive definite matrix.

According to Pontryagin's Maximum Principle (PMP), the optimal conditions for the above optimal control problem are [16]:

$$\begin{aligned} &\dot{x} = f(t, x) + g(t, x)[-R^{-1} g^T(t, x)\lambda] \\ &\dot{\lambda} = -\left(\frac{1}{2} \nabla Q(x) + (\frac{\partial f(t, x)}{\partial x})^T \lambda + \sum_{i=1}^{n} \lambda_i [-R^{-1} g^T(t, x)\lambda]^T \frac{\partial g_i(t, x)}{\partial x} \right) \\ &x(0) = x^0, \lambda(t_F) = 0, \end{aligned} \tag{2}$$

where $\lambda(t)$ is an n-element co-state vector and the optimal control law is given by:

$$u^* = -R^{-1} g^T(t, x)\lambda.$$

In general, (2) has no analytical solutions, thus some numerical methods should be applied. Shirazian et al. [16] solved this problem with the variational iteration method (VIM), which does not need any parametrizations and discretizations, but it suffers from time consuming analytical integrations. In [7], Jafari et al. made comparisons between the Adomian decomposition method (ADM), the homotopy perturbation method (HPM) and the VIM for solving this problem and showed that these methods yield similar solutions. Along with these semi-analytical approaches, a good idea of accelerating these methods by using pseudospectral routines was first introduced by Saberi et al. [13], where they combined the homotopy analysis method (HAM) with a Chebyshev pseudospectral method, which dramatically increased the convergence speed of the HAM [13]. Recently, Wang et al. have converted the nonlinear system (1) to a bilinear system, and then successively solve the modified optimality conditions, by the state-dependent Riccati equation method, taking the advantage of the Legendre pseudospectral method [18, 19].

Motivated by [13, 19], in this paper, we combine the SAM with the Legendre pseudospectral method, which preserves the convergence and speed advantage of pseudospectral methods in addition to solving a linear system of equations rather

than a nonlinear one. Strictly speaking, the pseudospectral procedure converts (2) to a system of nonlinear algebraic equations, which should be solved by some numerical methods such as the Newton method, as in [11], but our method only needs solving a linear system of algebraic equations. In comparison with the SAM, the VIM, the HAM, and their available modifications, our method does not need any integrations and this drastically reduces the CPU time. Finally, as an indirect method, our method benefits error control, in comparison with other direct methods, which is illustrated in numerical simulations.

Accordingly, the paper is organized as follows. In the first section, we review the SAM method for solving optimal conditions (2). Section 3 presents the proposed pseudospectral SAM. In Section 4, we prove the convergence of the method and analyze its error, and finally, in Section 5, we present the numerical simulations for some illustrative examples.

2. Pseudospectral SAM

For convenience, let us consider $X(t) = [X_1, X_2, \dots, X_{n+m}] := [x(t); \lambda(t)]$ and define the right-hand side of optimality conditions (2) as:

$$\Psi(t,x,\lambda) := \begin{bmatrix} f(t,x) + g(t,x)[-R^{-1}g^T(t,x)\lambda] \\ -\Big(\frac{1}{2}\nabla Q(x) + (\frac{\partial f(t,x)}{\partial x})^T\lambda + \sum_{i=1}^{n}\lambda_i[-R^{-1}g^T(t,x)\lambda]^T\frac{\partial g_i(t,x)}{\partial x}\Big) \end{bmatrix}.$$

Thus the TPBVP in (2) changes to the following compact form:

$$\begin{aligned} &\dot{X}(t) = \Psi(t, X(t)), \\ &X_{1:n}(0) = x^0, \quad X_{n+1:n+m}(t_F) = 0. \end{aligned}$$

We rewrite this equation in operator form as:

$$\begin{aligned} &\mathcal{L}_r[X(t)] + \mathcal{N}_r[X(t)] = 0, \quad r = 1, 2, \dots, n+m, \\ &X_{1:n}(0) = x^0, \quad X_{n+1:n+m}(t_F) = 0, \end{aligned} \tag{3}$$

where $\mathcal{L}_r$ and $\mathcal{N}_r$ are the linear and nonlinear parts of (2) defined as:

$$\begin{aligned} &\mathcal{L}_r[X(t)] = \dot{X}_r(t) + \sum_{i=1}^{n+m} p_{r,i}(t)X_i(t), \\ &\mathcal{N}_r[X(t)] = -\sum_{i=1}^{n+m} p_{r,i}(t)X_i(t) - \Psi_r(t, X(t)), \end{aligned} \tag{4}$$

where $\Psi_r(t, X(t))$ is the rth component of $\Psi(t, X(t))$. Now, we construct a sequence of solutions for solving (3), as follows:

$$\mathcal{L}_r[X_{k+1}(t)] = -\mathcal{N}_r[X_k(t)], \quad r = 1, 2, \dots, n+m, \tag{5}$$

for which $k \geq 0$, the first n entries of $X_{k+1}(t)$ at $t = 0$ are x^0 and its last n entries at $t = t_F$ are 0.

We explain the pseudospectral procedure for $t \in [0, t_F]$. Let $L_i(t)$ be the shifted Legendre polynomials generated by the recurrence relation:

$$L_{i+1}(t) = \frac{2i+1}{i+1}(\frac{2t}{t_F} - 1)L_i(t) - \frac{i}{i+1}L_{i-1}(t), \quad i = 1, 2, \dots$$

where $L_0(t) = 1$ and $L_1(t) = \frac{2t}{t_F} - 1$.

We denote by t_j^N, $0 \le j \le N$, Legendre-Gauss-Lobatto (LGL) points, which are defined by $t_0^N = 0$, $t_N^N = t_F$ and for $1 \le j \le N-1$, t_j^N the zeros of $\dot{L}_N(t)$, the derivative of the shifted Legendre polynomial of degree N.

Let $X_{r,k}(t) \in \mathbb{R}$ be the rth component of the unknown vector function $X_k(t)$. Then $X_{r,k}(t)$ can be approximated by means of the Legendre basis polynomials up to order N,

$$X_{r,k}(t) \approx X_{r,k}^N(t) = \sum_{j=0}^{N} L_j(t) X_{r,k}^{N,j},$$

where $X_{r,k}^{N,j}$ is the unknown coefficient of the Legendre polynomial of degree j, $L_j(t)$.

To approximate the derivatives of the unknown function $X_{r,k}(t)$ at the collocation points, we use the Legendre spectral differentiation matrix $\mathbf{D}$ as the matrix vector product

$$\dot{X}_{r,k}(t_j^N) \approx \dot{X}_{r,k}^N(t_j^N) = \sum_{i=0}^{N} \mathbf{D}_{ji} X_{r,k}^{N,j}, \quad j = 0, 1, \ldots, N,$$

or

$$\dot{X}_{r,k}(\mathbf{t}^N) \approx \dot{X}_{r,k}^N(\mathbf{t}^N) = \mathbf{D}\mathbf{Y}_{r,k}^N,$$

where $\mathbf{t}^N = [t_0^N, t_1^N, \ldots, t_N^N]^T$, and $\mathbf{Y}_{r,k}^N = [X_{r,k}^{N,0}, X_{r,k}^{N,1}, \ldots, X_{r,k}^{N,N}]^T$ is the vector of function $X_{r,k}(t)$ values at the collocation points and $\mathbf{D} = 2D/t_F$, where D is an $(N+1) \times (N+1)$ matrix whose entries are defined as [2]:

$$D_{ji} = \begin{cases} -\frac{N(N+1)}{4}, & \text{if } i = j = 0, \\ \frac{N(N+1)}{4}, & \text{if } i = j = N, \\ \frac{L_N(t_j^N)}{L_N(t_i^N)} \frac{t_F}{2(t_j^N - t_i^N)}, & \text{if } i \ne j, \\ 0, & \text{otherwise.} \end{cases}$$

Note that, in general, $\mathbf{D}^{(l)}$ is not equal to the differentiation matrix of order l.

Now, we substitute the approximate solution, $X_{r,k}^N(t)$, into (5) and require that it satisfies the equations at the LGL nodes. This requirement generates the following pseudospectral SAM (PSAM):

$$\mathbf{A}\mathbf{W}_{k+1}^N = -\mathbf{N}[\mathbf{W}_k^N], \quad k \ge 0, \tag{6}$$

$$\mathbf{W}_{k+1,1:n}^N(t_0^N) = x^0, \quad \mathbf{W}_{k+1,n+1:n+m}^N(t_N^N) = 0, \tag{7}$$

where

$$\mathbf{N}[\mathbf{W}_k^N] = [\mathbf{N}_1[\mathbf{W}_k^N]; \mathbf{N}_2[\mathbf{W}_k^N]; \ldots; \mathbf{N}_{n+m}[\mathbf{W}_k^N]]$$

is an $(N+1)(n+m)$ column vector whose $\mathbf{N}_r[\mathbf{W}_k^N]$ corresponds to $\mathcal{N}_r[X_{r,k}(t)]$ when evaluated at the collocation points for any $r = 1, 2, \ldots, N$, and $\mathbf{W}_k^N = [\mathbf{Y}_{1,k}^N; \mathbf{Y}_{2,k}^N; \ldots; \mathbf{Y}_{n+m,k}^N]$.

The matrix $\mathbf{A}$ is an $((N+1)(n+m))^2$ square block matrix derived from transforming the linear operators $\mathcal{L}_r$,$r = 1, \ldots, n+m$, at LGL collocation nodes, which

is defined as $\mathbf{A} = (A_{r,i})$,

$$A_{r,i} = \begin{cases} \mathbf{D} + p_{r,i}(\mathbf{t}^N)^T\mathbf{I}, & r = i, \\ p_{r,i}(\mathbf{t}^N)^T\mathbf{I}, & r \neq i, \end{cases}$$

where $\mathbf{I}$ is an identity matrix of order $N+1$.

To implement the boundary conditions (7), we replace the $(N+1)r$th rows of $A_{r,i}$ and $\mathbf{N}[\mathbf{W}_k^N]$, respectively, by the $(N+1)$th row of $\mathbf{I}$ and x_r^0, for $r = 1, 2, \ldots, n$, and rth rows of $A_{r,i}$ and $\mathbf{N}[\mathbf{W}_k^N]$ by the first row of $\mathbf{I}$ and 0, respectively, for $r = n+1, n+2, \ldots, n+m$. Then, at each step, the PSAM is related to solving a system of linear equations (6) to determine the unknown vector $\mathbf{W}_{k+1}^N$.

Remark 1. *In the dynamical system (1), if $x(t_F) = x_F$, then the boundary conditions in (3) become*

$$X_{1:n}(0) = x^0, \quad X_{1:n}(t_F) = x_F.$$

In this case, the PSAM boundary conditions (7) become

$$\boldsymbol{W}_{k+1,1:n}^N(t_0^N) = x^0, \quad \boldsymbol{W}_{k+1,1:n}^N(t_N^N) = x_F.$$

Therefore, to implement these boundary conditions, we should replace the $(N+1)r$th rows of $A_{r,i}$ and $\boldsymbol{N}[\boldsymbol{W}_k^N]$, respectively, by the $(N+1)$th row of $\boldsymbol{I}$ and x_r^0, and rth rows of $A_{r,i}$ and $\boldsymbol{N}[\boldsymbol{W}_k^N]$ by the first row of $\boldsymbol{I}$ and the rth element of x_F, respectively, for $r = 1, 2, \ldots, n$.

3. Convergence analysis

First, we recall the main steps of the proposed PSAM. To this end, we restate SAM (5) as:

$$\begin{aligned} &\mathcal{L}[X_{k+1}] = -\mathcal{N}[X_k], \\ &X_{k+1,1:n}(0) = x^0, \quad X_{k+1,n+1:n+m}(t_F) = 0, \end{aligned}$$

where $X_{r,k}$, $\mathcal{L}_r$ and $\mathcal{N}_r$ in (5) are the rth components of X_k and operators $\mathcal{L}$ and $\mathcal{N}$, respectively. Choose $\mathcal{L}[X] = \frac{d}{dt}X(t) + P(t)X(t)$ and $\mathcal{N}[X]$, the other remaining nonlinear components of (2) as in Section 2. Hence, the SAM can be rewritten as:

$$\begin{aligned} &\frac{d}{dt}X_{k+1}(t) + P(t)X_{k+1}(t) = -\mathcal{N}[X_k(t)], \\ &X_{k+1,1:n}(0) = x^0, \quad X_{k+1,n+1:n+m}(0) = \lambda^0, \end{aligned} \tag{8}$$

where $\lambda^0 \in \mathbb{R}^m$ is the root of $X_{K+1,n+1:n+m}^N(t_F, \lambda^0) = 0$, after some sufficient iterations of SAM (8), say K.

Approximating $X_k(t)$ by $X_k^N(t) \in (\mathcal{P}_{N+1}(0, t_F))^{n+m}$ and using the Legendre collocation method at any LGL point, t_j^N, results in PSAM formulation:

$$\begin{aligned} &\frac{d}{dt}X_{k+1}^N(t_j^N) + P(t_j^N)X_{k+1}^N(t_j^N) = -\mathcal{N}[X_k^N(t_j^N)], \quad \forall j = 1, \ldots, N, \\ &X_{k+1}^N(t_0^N) = \left[x^0; \lambda^0\right], \end{aligned} \tag{9}$$

where $\lambda^0 \in \mathbb{R}^m$ is the root of $X^N_{K+1,n+1:n+m}(t^N_N, \lambda^0) = 0$, after some sufficient iterations of PSAM (9), K.

In this section, we will use this formulation which is equivalent to (6) because of its straightforward convergence analysis.

Definition 1. *Let $(u,v)_{t_F}$ and $\|v\|_{t_F}$ be the inner product and the norm of space $\big(L^2(0,t_F)\big)^{n+m}$, respectively. We define the following discrete inner product and norm,*

$$(u,v)_{t_F,N} = \sum_{j=0}^{N} u^T(t^N_j)v(t^N_j)\omega^N_j, \qquad \|v\|_{t_F,N} = (v,v)^{\frac{1}{2}}_{t_F,N},$$

where ω^N_j are the Christoffel numbers corresponding to the LGL points t^N_j, $0 \le j \le N$.

Two important properties of Legendre Guass-Lobatto quadratures are:

1. For any $L\psi \in \mathcal{P}_{2N+1}(0,t_F)$,

$$(L,\psi)_{t_F} = (L,\psi)_{t_F,N}.$$

2. The norm and the discrete norm are equivalent, *i.e.*

$$\|L\|_{t_F} \le \|L\|_{t_F,N} \le \sqrt{2+\frac{1}{N}}\|L\|_{t_F}. \tag{10}$$

Theorem 1. *Assume that for any $j = 0,1,\dots,N$, $\mathcal{X}_j = \{X^N_{k+1}(t^N_j), k = 0,1,\dots\}$ is the PSAM sequence produced by (9). Furthermore, assume*

$$\alpha = \max\{\|P(t^N_j)\|, j = 0,1,\dots,N\},$$

and

$$\|\Psi(.,X^N_k) - \Psi(.,X^N_{k-1})\|_{t_F,N} \le M\|X^N_k - X^N_{k-1}\|_{t_F,N},$$

for some constant $M > 0$. Then for any initial n-vector $X^N_0(t^N_j)$, $\mathcal{X}_j$ converges to some $\hat{X}(t^N_j)$ which is the exact solution of (3) at any LGL point, t^N_j, if

$$\frac{4t_F\sqrt{2+\frac{1}{N}}(\alpha+M)}{1-4\alpha t_F\sqrt{2+\frac{1}{N}}} \le \beta < 1. \tag{11}$$

Moreover, $\hat{X}^N(t) := \sum_{j=0}^N \phi_j(t)\hat{X}(t^N_j)$ is an analytic approximate solution of (3), which gives the exact solutions at any LGL point $t^N_j, j = 0,1,\dots,N$.

Proof. We assume that $k \ge 1$ and $1 \le r \le n+m$, throughout the proof. Define $\tilde{X}^N_{k+1} := X^N_{k+1} - X^N_k$. By (9), we have

$$\begin{aligned}\frac{d}{dt}\tilde{X}^N_{k+1}(t^N_j) + P(t^N_j)\tilde{X}^N_{k+1}(t^N_j) = {} & P(t^N_j)\tilde{X}^N_k(t^N_j) \\ & +\Psi(t^N_j, X^N_k(t^N_j)) - \Psi(t^N_j, X^N_{k-1}(t^N_j)),\end{aligned}$$

which results in

$$\frac{d}{dt}\tilde{X}^N_{k+1}(t^N_j) = -P(t^N_j)\left(\tilde{X}^N_{k+1}(t^N_j) - \tilde{X}^N_k(t^N_j)\right) + \Psi(t^N_j, X^N_k(t^N_j)) - \Psi(t^N_j, X^N_{k-1}(t^N_j)). \tag{12}$$

Obviously, $\tilde{X}^N_{k+1}(0) = 0$ and

$$\tilde{X}^N_{k+1}(t_F)^T\tilde{X}^N_{k+1}(t_F) = 2\left(\tilde{X}^N_{k+1}, \frac{d}{dt}\tilde{X}^N_{k+1}\right)_{t_F} \le 2\|\tilde{X}^N_{k+1}\|_{t_F}\|\frac{d}{dt}\tilde{X}^N_{k+1}\|_{t_F}. \tag{13}$$

Moreover, for any $t \in [0, t_F]$,

$$\begin{aligned}\tilde{X}^N_{k+1}(t)^T\tilde{X}^N_{k+1}(t) &= \tilde{X}^N_{k+1}(t_F)^T\tilde{X}^N_{k+1}(t_F) - \int_t^{t_F}\frac{d}{ds}\left(\tilde{X}^N_{k+1}(s)^T\tilde{X}^N_{k+1}(s)\right)ds \\ &\le \tilde{X}^N_{k+1}(t_F)^T\tilde{X}^N_{k+1}(t_F) + 2\|\tilde{X}^N_{k+1}\|_{t_F}\|\frac{d}{dt}\tilde{X}^N_{k+1}\|_{t_F}.\end{aligned}$$

Then integrating with respect to t yields

$$\|\tilde{X}^N_{k+1}\|^2_{t_F} \le t_F\tilde{X}^N_{k+1}(t_F)^T\tilde{X}^N_{k+1}(t_F) + 2t_F\|\tilde{X}^N_{k+1}\|_{t_F}\|\frac{d}{dt}\tilde{X}^N_{k+1}\|_{t_F},$$

from where

$$\tilde{X}^N_{k+1}(t_F)^T\tilde{X}^N_{k+1}(t_F) \ge \frac{1}{t_F}\tilde{X}^N_{k+1}(t_F)^T\tilde{X}^N_{k+1}(t_F) - 2t_F\|\tilde{X}^N_{k+1}\|_{t_F}\|\frac{d}{dt}\tilde{X}^N_{k+1}\|_{t_F}. \tag{14}$$

Using (13) and (14), we get

$$\|\tilde{X}^N_{k+1}\|_{t_F} \le 4t_F\|\frac{d}{dt}\tilde{X}^N_{k+1}\|_{t_F}. \tag{15}$$

On the other hand, by the definition of α and applying (12), we can write

$$\begin{aligned}\|\frac{d}{dt}\tilde{X}^N_{k+1}\|_{t_F} &\le \|\frac{d}{dt}\tilde{X}^N_{k+1}\|_{t_F,N} \\ &\le \|P(t^N_j)\|\left(\|\tilde{X}^N_{k+1}\|_{t_F,N} + \|\tilde{X}^N_k\|_{t_F,N}\right) + M\|\tilde{X}^N_k\|_{t_F,N} \\ &\le \alpha\|\tilde{X}^N_{k+1}\|_{t_F,N} + (\alpha+M)\|\tilde{X}^N_k\|_{t_F,N},\end{aligned} \tag{16}$$

By property (10), inequalities (15) and (16) yield

$$\begin{aligned}\|\tilde{X}^N_{k+1}\|_{t_F} &\le 4t_F\|\frac{d}{dt}\tilde{X}^N_{k+1}\|_{t_F} \\ &\le 4t_F\sqrt{2+\frac{1}{N}}\left\{\alpha\|\tilde{X}^N_{k+1}\|_{t_F} + (\alpha+M)\|\tilde{X}^N_k\|_{t_F}\right\}.\end{aligned}$$

Therefore

$$\|\tilde{X}^N_{k+1}\|_{t_F} \le \frac{4t_F\sqrt{2+\frac{1}{N}}(\alpha+M)}{1-4\alpha t_F\sqrt{2+\frac{1}{N}}}\|\tilde{X}^N_k\|_{t_F} \le \beta\|\tilde{X}^N_k\|_{t_F}.$$

Hence, we have

$$\|\tilde{X}_{k+1}^N\|_{t_F} \leq \beta \|\tilde{X}_k^N\|_{t_F} \leq \cdots \leq \beta^k \|\tilde{X}_1^N\|_{t_F}$$

Then for any $k' \geq k \geq 1$,

$$\begin{aligned}\|X_{k'}^N - X_k^N\|_{t_F} &\leq \sum_{i=k}^{k'-1} \|\tilde{X}_{i+1}^N\|_{t_F} \leq \sum_{i=k}^{k'-1} \beta^i \|\tilde{X}_1^N\|_{t_F} \\ &\leq \sum_{i=k}^{\infty} \beta^i \|\tilde{X}_1^N\|_{t_F} \leq \frac{\beta^k}{1-\beta} \|\tilde{X}_1^N\|_{t_F}.\end{aligned}$$

Since $\beta \in [0,1)$, $\|X_{k'}^N - X_k^N\|_{t_F} \to 0$ as $k, k' \to \infty$. Thus $\mathcal{X}_j$ is a Cauchy sequence, and since $\mathbb{R}^n$ is a Banach space, $\mathcal{X}_j$ has a limit $\hat{X}(t_j^N)$. Taking limit $k \to \infty$ in (9) yields

$$\begin{aligned}&\frac{d}{dt}\hat{X}(t_j^N) + P(t_j^N)\hat{X}(t_j^N) = -\mathcal{N}[\hat{X}(t_j^N)], \quad \forall j = 1, \ldots, N, \\ &\hat{X}(t_0^N) = \left[x^0; \lambda^0\right].\end{aligned}$$

Thus, $\hat{X}(t_j^N)$ is the exact solution of (3) at any LGL point t_j^N. In addition, due to the definition of $\hat{X}^N(t)$, it is easy to verify that $\hat{X}^N(t_j^N) = \hat{X}(t_j^N)$ and the proof is complete. □

Corollary 1. *If $\|\tilde{X}_1^N\|_{t_F} \leq \delta$, then PSAM needs $k \geq \left[\frac{\log(\epsilon/\delta)}{\log \beta}\right] + 1$ iterations to reach a given tolerance limit $\epsilon > 0$.*

Corollary 2. *The PSAM convergence criterion (11) is equivalent to $\alpha \leq \frac{\beta - 4t_F M\sqrt{2+\frac{1}{N}}}{4t_F\sqrt{2+\frac{1}{N}}(1+\beta)}$, where $\beta < 1$. One can use this equivalence to check the appropriacy of choosing $P(t)$.*

Corollary 3. *Assume that $\beta_1, \beta_2 \in [0,1)$ are two convergence constants in (11), for different choices of $P(t)$, say $P_1(t)$ and $P_2(t)$. Then $\beta_1 < \beta_2$ implies that the convergence rate of the first PSAM is more than the second one.*

Theorem 2. *Under the assumptions of Theorem 1, the sequences $\{u_k^N(t_j^N), k = 0, 1, \ldots\}$ and $\{J_k^N(t_j^N), k = 0, 1, \ldots\}$ defined by*

$$u_{k,j}^N = -R^{-1} g^T(t_j^N, X_{k,1:n}^N(t_j^N)) X_{k,n+1:n+m}^N(t_j^N) \tag{17}$$

$$J_k^N = \frac{T}{4} \sum_{j=0}^{N} w_j \left\{ Q\left(X_{k,1:n}^N(t_j^N)\right) + (u_{k,j}^N)^T R u_{k,j}^N \right\} \tag{18}$$

converges to optimal control values at LGL points, say $\hat{u}_j^N$, and optimal objective value, $\hat{J}^N$. Accordingly, the optimal control law is obtained by $\hat{u}^N(t) = \sum_{j=0}^N L_j(t)\hat{u}_j^N$.

Proof. First, note that $[X^N_{k,1:n}; X^N_{k,n+1:n+m}] = X^N_k$. According to Theorem 1, the convergence of $\{X^N_k(t^N_j), k = 0, 1, \ldots\}$ results in the convergence of $\{X^N_{k,1:n}(t^N_j), k = 0, 1, \ldots\}$ and $\{X^N_{k,n+1:n+m}(t^N_j), k = 0, 1, \ldots\}$ to some vectors $\hat{X}^N_{1:n}(t^N_j)$ and $\hat{X}^N_{n+1:n+m}(t^N_j)$. By taking the limit from (17), the continuity assumption of $g(t, x)$ gives

$$\begin{aligned}\hat{u}^N_j := \lim_{k\to\infty} u^N_{k,j} &= -R^{-1}g^T(t^N_j, \lim_{k\to\infty} X^N_{k,1:n}(t^N_j))(\lim_{k\to\infty} X^N_{k,n+1:n+m}(t^N_j)) \\ &= -R^{-1}g^T(t^N_j, \hat{X}^N_{1:n}(t^N_j))\hat{X}^N_{n+1:n+m}(t^N_j),\end{aligned}$$

which is the optimal control law since $\hat{X}^N_{1:n}(t^N_j)$ and $\hat{X}^N_{n+1:n+m}(t^N_j)$ are the optimal state and costate vectors. Hence, taking the limit from (18), as $k \to \infty$, since N is a finite constant and $Q(.)$ is a continuous function, we have that

$$\begin{aligned}\hat{J}^N := \lim_{k\to\infty} J^N_k &= \frac{T}{4}\sum_{j=0}^{N} w_j \left\{ Q\left(\lim_{k\to\infty} X^N_{k,1:n}(t^N_j)\right) + (\lim_{k\to\infty} u^N_{k,j})^T R \lim_{k\to\infty} u^N_{k,j} \right\} \\ &= \frac{T}{4}\sum_{j=0}^{N} w_j \left\{ Q\left(\hat{X}^N_{1:n}(t^N_j)\right) + (\hat{u}^N_j)^T R \hat{u}^N_j \right\},\end{aligned}$$

and this ends the proof, as $\hat{X}^N_{1:n}(t^N_j)$ and $\hat{u}^N_j$ are the optimal state and control vectors. □

4. Illustrative example

In this section, three examples are given to illustrate the simplicity and efficiency of the proposed method. The codes are developed using computation software MATLAB 7.12.0, and the calculations are implemented on a machine with Intel Core 2 Due Processor 2.53 Ghz with 4 GB of RAM.

Example 1. *Consider a two-dimensional nonlinear composite system described by*

$$\begin{aligned}\dot{x}_1 &= x_1 - x_1^3 + x_2^2 + u_1 \\ \dot{x}_2 &= -x_2 + x_2(x_1 + x_2^2) + u_2 \\ x_1(0) = 0, &\quad x_2(0) = 0.8.\end{aligned}$$

The quadratic cost functional to be minimized is given by:

$$J = \frac{1}{2}\int_0^1 (x_1^2 + x_2^2 + u_1^2 + u_2^2)dt.$$

The optimality conditions are

$$\begin{aligned}\dot{x}_1 &= x_1 - x_1^3 + x_2^2 - \lambda_1 \\ \dot{x}_2 &= -x_2 + x_2(x_1 + x_2^2) - \lambda_2 \\ \dot{\lambda}_1 &= -(x_1 + \lambda_1(1 - 3x_1^2) + \lambda_2 x_2) \\ \dot{\lambda}_2 &= -(x_2 + 2\lambda_1 x_2 + \lambda_2(-1 + x_1 + 3x_2^2)) \\ x_1(0) = 0, &\quad x_2(0) = 0.8, \quad \lambda_1(1) = 0, \quad \lambda_2(1) = 0,\end{aligned}$$

and the optimal control laws are given by $u_1^* = -\lambda_1$ *and* $u_2^* = -\lambda_2$.

In view of (4), consider the linear operator $\mathcal{L}[X] = \dot{X}(t) + P(t)X(t)$, *and the nonlinear operator* $\mathcal{N}[X]$ *as follows:*

$$P(t) = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 \end{bmatrix}, \ \mathcal{N}[X] = \begin{bmatrix} -x_1^3 + x_2^2 \\ x_2(x_1 + x_2^2) \\ 3\lambda_1 x_1^2 - \lambda_2 x_2 \\ -(2\lambda_1 x_2 + \lambda_2(x_1 + 3x_2^2)) \end{bmatrix},$$

where $X = [x_1, x_2, \lambda_1, \lambda_2]^T$.

k	CPU time (sec.)	Max error PSAM	CPU time (sec.)	Max error VIM	CPU time (sec.)	Max error HAM
2	0.00079	6.4322e-2	0.047	5.1463e-1	0.34972	6.5807e-2
3	0.00119	1.8859e-2	0.094	1.7670e-1	1.19685	6.0841e-2
4	0.00311	4.1203e-3	0.109	1.3528e-1	3.05910	5.2627e-2
5	0.00335	9.5764e-4	0.281	5.6512e-2	7.99916	4.0040e-2
6	0.00367	1.5836e-4	1.918	1.1235e-2	18.64958	2.7282e-2

Table 1: *The maximum error of PSAM for* $x_1(t)$ *with* $N = 20$, *compared to VIM [16] and HAM [5]*

Method	Objective value	CPU time (sec.)
PSAM ($k = 6$, $N = 20$)	0.18887	0.00367
bvp4c	0.18890	0.10655
VIM ($k = 6$)	0.18817	1.918
HAM ($k = 6$, $h = -0.45$)	0.18832	18.64958

Table 2: *Comparison of objective value,* J, *for PSAM and other methods, Example 1*

Implementing PSAM (6)-(7), the suboptimal solutions are obtained for $N = 20$. *To analyze the accuracy and efficiency of the proposed method, we have also solved the problem by two methods, VIM [16] and HAM [5]. Table 1 shows better accuracy of the PSAM while preserving the CPU time. Precisely, as the number of iteration* k *increases, the maximum errors of the PSAM vanish more faster and in less CPU time compared to the VIM and the HAM.*

Moreover, according to Table 2, the suboptimal objective value of the proposed PSAM is more consistent with the MATLAB built-in function, bvp4c *[8], which ensures better accuracy of our obtained solution compared to VIM [16] and HAM [5].*

The suboptimal states and controls of the PSAM, the VIM and the HAM for $k = 6$ *and* bvp4c *are depicted in figures 1-2.*

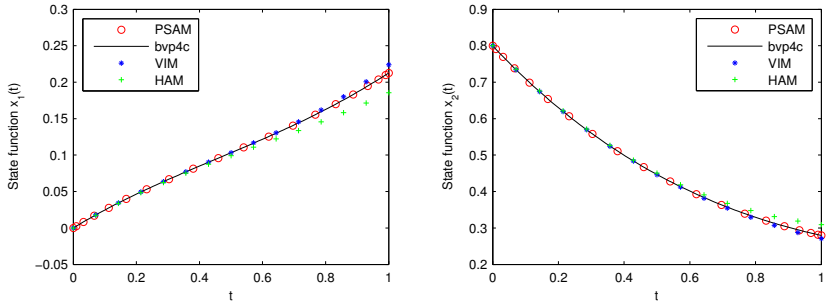

Figure 1: *Suboptimal states, Example 1*

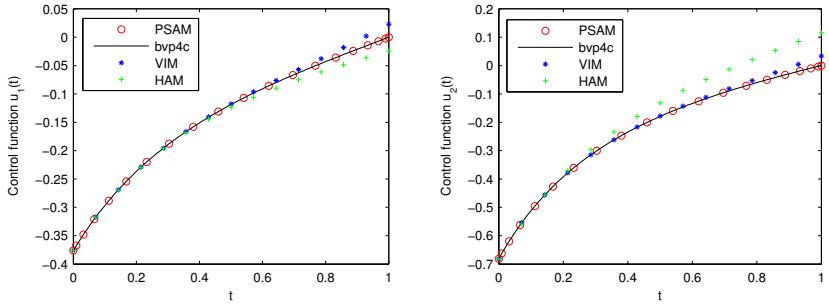

Figure 2: *Suboptimal controls, Example 1*

Example 2. *Consider the following optimal control problem for the Van Der Pol oscillator [4]:*

$$\begin{aligned} \textit{minimize} \quad & J = \frac{1}{2}\int_0^1 (x_1^2 + x_2^2 + u^2)dt \\ \textit{subject to: } & \dot{x}_1 = x_2 \\ & \dot{x}_2 = -x_1 + x_2(1 - x_1^2) + u \\ & x_1(0) = 1, \quad x_2(0) = 0. \end{aligned}$$

The extreme conditions will be:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 + x_2(1 - x_1^2) - \lambda_2 \\ \dot{\lambda}_1 &= -x_1 + \lambda_2(1 + 2x_1x_2) \\ \dot{\lambda}_2 &= -x_2 - \lambda_1 - \lambda_2(1 - x_1^2) \\ x_1(0) = 1, \quad & x_2(0) = 0, \quad \lambda_1(1) = \lambda_2(1) = 0. \end{aligned}$$

The optimal control law is also given by

$$u^*(t) = -\lambda_2(t), \quad t \in [0, 1].$$

k	CPU time (sec.)	Max error PSAM	CPU time (sec.)	Max error SHAM (Legendre)	CPU time (sec.)	Max error DTM
8	0.0170	2.1962e-4	0.101	6.7082e-3	7.802	3.2102e-4
10	0.0189	6.4172e-5	0.118	8.3837e-3	14.648	3.9828e-4
14	0.0252	8.9001e-7	0.177	1.9506e-3	47.815	4.4387e-4
15	0.0280	8.8769e-7	0.200	4.2749e-4	87.745	4.4380e-4

Table 3: *The maximum error of the PSAM for $x_1(t)$ with $N = 50$ compared to SHAM [13] and DTM [4], Example 2*

Method	Objective value	CPU time (sec.)
PSAM ($k=15$, $N=50$)	1.047806908	0.0279
bvp4c	1.047806895	0.0649
SHAM Legendre ($k=15$, $N=50$, $h=-0.5$)	1.0472	0.188
DTM ($k = 15$)	1.0478	87.745

Table 4: *Comparison of objective function value, J, for the PSAM and other methods, Example 2*

In view of (4), consider the linear operator $\mathcal{L}[X] = \dot{X}(t)$, and the other terms as the nonlinear operator $\mathcal{N}[X]$, where $X = [x_1, x_2, \lambda_1, \lambda_2]^T$. Using PSAM (6)-(7), we have obtained the suboptimal solutions for $N = 50$.

For analyzing the accuracy and efficiency of our proposed method, we compare the PSAM with two recent methods, SHAM [13] and DTM [4]. The maximum errors of the PSAM, the SHAM and the DTM are summarized in Table 3. It verifies better accuracy of the PSAM, while maintaining the CPU time. In addition, as the number of iteration k increases, the maximum errors of the PSAM vanish rapidly, in less than 0.03 seconds, while the SHAM and the DTM do not.

Furthermore, Table 4 presents a comparison of objective values of the PSAM, bvp4c, the SHAM and the DTM, which ensures better accuracy of the proposed PSAM.

Figures 3 and 4 show suboptimal states and control for $k = 15$ iterations of the PSAM, compared to the SHAM and MATLAB built-in function bvp4c.

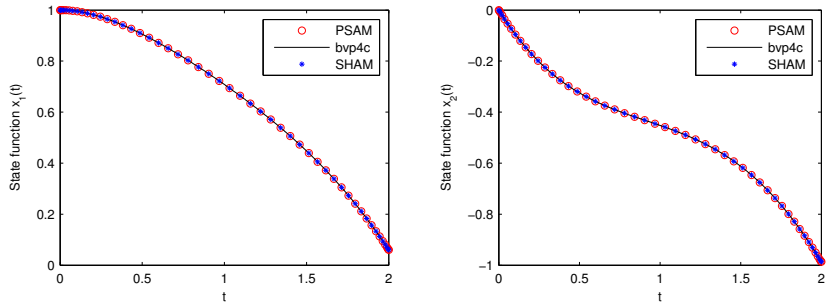

Figure 3: *Suboptimal states, Example 2*

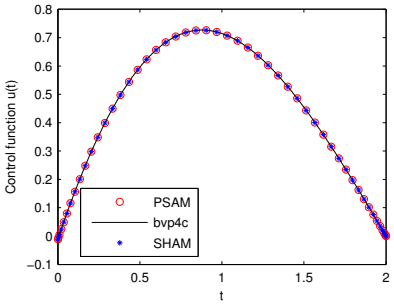

Figure 4: *Suboptimal control, Example 2*

Example 3. *Consider the optimal maneuvers of a rigid asymmetric spacecraft [13].*
The Euler equations for the angular velocities of the spacecraft are given by:

$$\dot{x}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} -\frac{I_3 - I_2}{I_1} x_2(t)x_3(t) \\ -\frac{I_1 - I_3}{I_2} x_1(t)x_3(t) \\ -\frac{I_2 - I_1}{I_3} x_1(t)x_2(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{I_1} & 0 & 0 \\ 0 & \frac{1}{I_2} & 0 \\ 0 & 0 & \frac{1}{I_3} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix},$$

where x_1, x_2, and x_3 are angular velocities of the spacecraft, u_1, u_2, and u_3 are control torques, $I_1 = 86.24$, $I_2 = 85.07$, and $I_3 = 113.59$ $kg\ m^2$ are the spacecraft principle inertia.
The quadratic cost functional to be minimized is given by:

$$J[x,u] = \frac{1}{2}\int_0^{100} (x^T(t)Qx(t) + u^T(t)Ru(t))dt,$$

where $Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

In addition, the following boundary conditions should be satisfied:

$$\begin{cases} x_1(0) = 0.01\,r/s, \quad x_2(0) = 0.005\,r/s, \quad x_3(0) = 0.001\,r/s, \\ x_1(100) = x_2(100) = x_3(100) = 0\,r/s. \end{cases}$$

According to the PMP, the extreme conditions form the following nonlinear TPBVP:

$$\begin{aligned}
\dot{x}_1(t) &= -\frac{\lambda_1(t)}{I_1^2} - \frac{I_3 - I_2}{I_1} x_2(t)x_3(t), \\
\dot{x}_2(t) &= -\frac{\lambda_2(t)}{I_2^2} - \frac{I_1 - I_3}{I_2} x_1(t)x_3(t), \\
\dot{x}_3(t) &= -\frac{\lambda_3(t)}{I_3^2} - \frac{I_2 - I_1}{I_3} x_1(t)x_2(t), \\
\dot{\lambda}_1(t) &= \frac{I_1 - I_3}{I_2} x_3(t)\lambda_2(t) + \frac{I_2 - I_1}{I_3} x_2(t)\lambda_3(t), \\
\dot{\lambda}_2(t) &= \frac{I_3 - I_2}{I_1} x_3(t)\lambda_1(t) + \frac{I_2 - I_1}{I_3} x_1(t)\lambda_3(t),
\end{aligned}$$

$$\dot{\lambda}_3(t) = \frac{I_3 - I_2}{I_1}\, x_2(t)\lambda_1(t) + \frac{I_1 - I_3}{I_2}\, x_1(t)\lambda_2(t),$$
$$x_1(0) = 0.01\, r/s, \quad x_2(0) = 0.005\, r/s, \quad x_3(0) = 0.001\, r/s,$$
$$x_1(100) = x_2(100) = x_3(100) = 0\, r/s,$$

and the optimal control laws are given by:

$$u_1^*(t) = -\frac{\lambda_1(t)}{I_1}, \quad t \in [0, 100],$$
$$u_2^*(t) = -\frac{\lambda_2(t)}{I_2}, \quad t \in [0, 100],$$
$$u_3^*(t) = -\frac{\lambda_3(t)}{I_3}, \quad t \in [0, 100].$$

In order to apply PSAM (6), we should apply boundary conditions (8) discussed in Remark (1). Let $X = [x_1, x_2, x_3, \lambda_1, \lambda_2, \lambda_3]^T$, *and*

$$p_{14}(t) = \frac{1}{I_1^2}, \quad p_{25}(t) = \frac{1}{I_2^2}, \quad p_{36} = \frac{1}{I_3^2},$$

and $p_{i,j}(t) = 0$, *otherwise. For this choice of* $P(t)$, *the convergence constant in (11) is* $\beta = 0.1514 \in [0, 1)$, *which indicates the convergence of the PSAM. Table 5 shows good accuracy and better CPU times of our proposed method, in contrast to SHAM [13] and DTM [4]. In addition, Table 6 verifies the better accuracy of the PSAM suboptimal objective value, compared to the SHAM and the HAM [5].*

Suboptimal states and controls are depicted in figures 5-7 for $k = 6$ *iterations of the PSAM, in comparison with the SHAM and* bvp4c.

Our final discussion is about the choice of the linear operator $\mathcal{L}[X(t)] = \dot{X}(t) + P(t)X(t)$, *or equivalently, the choice of* $P(t)$. *For this example, a trivial alternative choice of* $P(t)$ *is the zero constant matrix, for which* $\beta = 0.1510 \in [0, 1)$. *This guarantees the convergence of* $P(t)$ *by Theorem 1. We have compared the results of these two choices of* $P(t)$ *in Table 7. One can easily see better accuracy of the first case in contrast with the second one. The CPU times of these two cases of the PSAM are less than* 0.035 *second, which shows excellent speeds for both cases.*

k	CPU time (sec.)	Max error PSAM	CPU time (sec.)	Max error SHAM (Legendre)	CPU time (sec.)	Max error DTM
2	0.0126	2.1471e-6	0.079	1.0206e-6	7.328	7.3102e-5
3	0.0169	9.9853e-9	0.118	1.8130e-7	12.542	9.2612e-6
4	0.0197	5.6034e-8	0.155	3.2309e-8	18.664	1.9301e-6
5	0.0230	6.8025e-9	0.198	5.7802e-9	36.729	2.2560e-7
6	0.0273	6.3763e-9	0.224	1.0589e-9	46.401	3.1420e-8

Table 5: *The maximum error of the PSAM for* $x_1(t)$ *with* $N = 50$, *compared to SHAM [13] and DTM [4], Example 3*

Method	Objective value	CPU time (sec.)
PSAM (k=6, N=50)	0.004687796532	0.0265
bvp4c	0.004687799751	0.0389
SHAM Legendre (k=6, N=50, h=-1.2)	0.004687794463	0.227
HAM (k=3, h=-1)	0.004687795533	10.821

Table 6: *Comparison of objective function value, J, for the PSAM and other methods, Example 3*

k	CPU time (sec.)	Max error PSAM (case I)	CPU time (sec.)	Max error PSAM (case II)
1	0.00366	4.99220e-3	0.00397	1.16410e-2
2	0.00884	3.33250e-4	0.00751	3.16210e-3
3	0.01247	1.32100e-4	0.01252	2.09870e-3
4	0.01596	1.02420e-5	0.01622	1.99420e-4
5	0.01929	4.30480e-6	0.01993	1.60540e-4
6	0.02370	1.81510e-6	0.02343	7.06480e-5
7	0.02722	1.68970e-6	0.02701	2.66670e-5
8	0.03055	1.69100e-6	0.03046	3.11730e-6
9	0.03386	1.68850e-6	0.03405	7.18870e-7
10	0.03717	1.68860e-6	0.03754	1.68950e-6

Table 7: *The maximum error of the PSAM for two different choices of $P(t)$ for $N = 50$, Example 3*

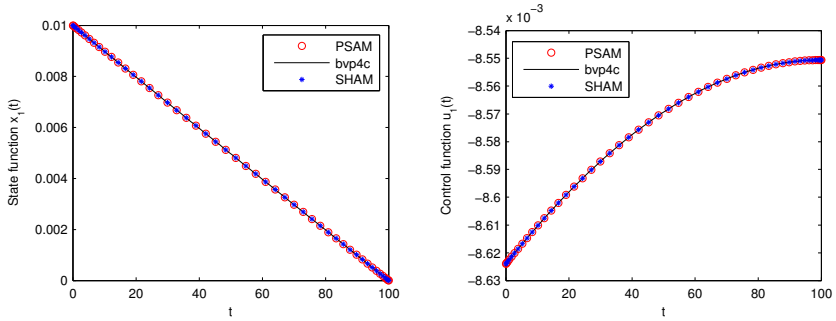

Figure 5: *Suboptimal control and state $x_1(t)$ and $u_1(t)$, Example 3*

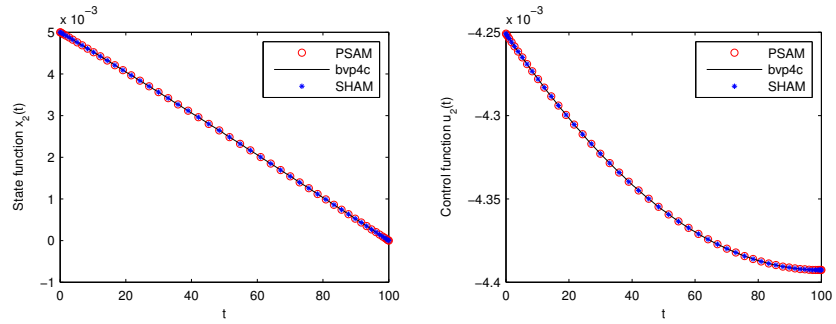

Figure 6: *Suboptimal control and state $x_2(t)$ and $u_2(t)$, Example 3*

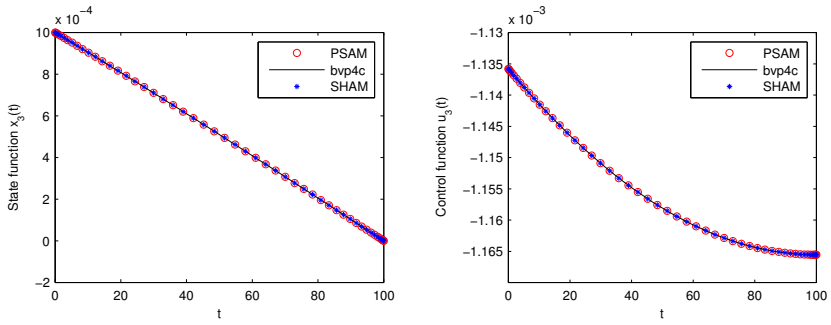

Figure 7: *Suboptimal control and state $x_3(t)$ and $u_3(t)$, Example 3*

5. Conclusions

In this paper, a pseudospectral successive approximation method is proposed for solving a broad class of optimal control problems. This method can solve the TP-BVP obtained from PMP in a recursive manner. The proposed PSAM does not need any complex computations in comparison with semi-analytical methods, such as the VIM and the HPM. In comparison with recent direct pseudospectral methods, it benefits solving a sequence of linear algebraic equations instead of the nonlinear system. It also preserves fast convergence of pseudospectral methods. The convergence of the proposed PSAM is proved and two illustrative examples demonstrated the effectiveness and good results in short CPU time.

Acknowledgement

The author would like to give his special thanks to professor Hassan Saberi Nik for his insightful comments on proving the convergence theorem of the PSAM, Theorem 1.

References

[1] M. Alipour, M. A. Vali, A. H. Borzabadi, *A hybrid parametrization approach for a class of nonlinear optimal control problems*, Numer. Algebra Control Optim. **9**(2019), 493.

[2] C. Canuto, M. Yousuff Hussaini, A. Quarteroni, A. Thomas Jr, et al., *Spectral methods in fluid dynamics*, Springer Science & Business Media, Berlin, 2012.

[3] B. Chen-Charpentier, M. Jackson, *Optimal control of plant virus propagation*, Math. Methods Appl. Sci. **43**(2020), 8147–8157.

[4] D. Du, I. Hwang, *A computational approach to solve optimal control problems using differential transformation*, in: *2007 American Control Conference*, IEEE, New York, 2007, 2322–2327.

[5] S. Effati, H. Saberi Nik, M. Shirazian, *Analytic-approximate solution for a class of nonlinear optimal control problems by homotopy analysis method*, Asian-Eur. J. Math. **6**(2031), 1350012.

[6] X. Gao, K. L. Teo, G. -R. Duan, *An optimal control approach to spacecraft rendezvous on elliptical orbit*, Optim. Control Appl. Methods **36**(2015), 158–178.

[7] H. Jafari, S. Ghasempour, D. Baleanu, *On comparison between iterative methods for solving nonlinear optimal control problems*, J. Vib. Control **22**(2016), 2281–2287.
[8] J. Kierzenka, L. F. Shampine, *A bvp solver based on residual control and the maltab pse*, ACM Trans. Math. Softw. (TOMS) **27**(2001), 299–316.
[9] S. Lenhart, J. T. Workman, *Optimal control applied to biological models*, Chapman and Hall/CRC, London, 2007.
[10] T. Li, Y. Guo, *Nonlinear dynamical analysis and optimal control strategies for a new rumor spreading model with comprehensive interventions*, Qual. Theory Dyn. Syst. **20**(2021), 1–24.
[11] M. A. Mehrpouya, H. Peng, *A robust pseudospectral method for numerical solution of nonlinear optimal control problems*, Int. J. Comput. Math. **98**(2021), 1146–1165.
[12] H. Mirinejad, T. Inanc, *An rbf collocation method for solving optimal control problems*, Rob. Auton. Syst. **87**(2017), 219–225.
[13] H. Saberi Nik, S. Effati, S. S. Motsa, M. Shirazian, *Spectral homotopy analysis method and its convergence for solving a class of nonlinear optimal control problems*, Numer. Algorithms **65**(2014), 171–194.
[14] S. Saha, A. Banerjee, S. M. Amrr, M. Nabi, *Pseudospectral method-based optimal control for a nonlinear five degree of freedom active magnetic bearing system*, Trans. Inst. Meas. Control. **43**(2021), 1668–1679.
[15] M. Salik, A. Banerjee, M. Nabi, *Pseudospectral method based optimal control of tuberculosis model*, in: *2020 28th Mediterranean Conference on Control and Automation (MED)*, IEEE, Saint-Raphaël, 2020, 899–904.
[16] M. Shirazian, S. Effati, *Solving a class of nonlinear optimal control problems via hes variational iteration method*, Int. J. Control Autom. Syst. **10**(2012), 249–256.
[17] X. Tang, H. Xu, *Multiple-interval pseudospectral approximation for nonlinear optimal control problems with time-varying delays*, Appl. Math. Model. **68**(2021), 137–151.
[18] Z. Wang, Y. Li, *Linear direct transcription for nonlinear constrained optimal control problems*, IFAC-PapersOnLine **53**(2020), 6981–6986.
[19] Z. Wang, Y. Li, *State-dependent indirect pseudospectral method for nonlinear optimal control problems*, ISA Trans. **108**(2021), 220–229.
[20] L. Zhu, M. Liu, Y. Li, *The dynamics analysis of a rumor propagation model in online social networks*, Phys. A: Stat. Mech. Appl. **520**(2019), 118–137.
[21] L. Zhu, X. Zhou, Y. Li, *Global dynamics analysis and control of a rumor spreading model in online social networks*, Phys. A: Stat. Mech. Appl. **526**(2019), 120903.