Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Galerkin approximation with Legendre polynomials for a continuous-time nonlinear optimal control problem*

Xue-song CHEN

(*School of Applied Mathematics, Guangdong University of Technology, Guangzhou 510006, China*)

E-mail: chenxs@gdut.edu.cn

Received June 4, 2016; Revision accepted Oct. 15, 2016; Crosschecked Nov. 1, 2017

Abstract: We investigate the use of an approximation method for obtaining near-optimal solutions to a kind of nonlinear continuous-time (CT) system. The approach derived from the Galerkin approximation is used to solve the generalized Hamilton-Jacobi-Bellman (GHJB) equations. The Galerkin approximation with Legendre polynomials (GALP) for GHJB equations has not been applied to nonlinear CT systems. The proposed GALP method solves the GHJB equations in CT systems on some well-defined region of attraction. The integrals that need to be computed are much fewer due to the orthogonal properties of Legendre polynomials, which is a significant advantage of this approach. The stabilization and convergence properties with regard to the iterative variable have been proved. Numerical examples show that the update control laws converge to the optimal control for nonlinear CT systems.

Key words: Generalized Hamilton-Jacobi-Bellman equation; Nonlinear optimal control; Galerkin approximation; Legendre polynomials

https://doi.org/10.1631/FITEE.1601101 **CLC number:** TP13

1 Introduction

The optimal feedback control of nonlinear systems is a challenging problem. The dynamic programming algorithm is widely regarded as the most effective method for obtaining optimal feedback control laws for some nonlinear systems. Recently, Yu and Jiang (2015) developed some novel adaptive dynamic programming to achieve global and adaptive suboptimal stabilization of uncertain continuous-time nonlinear polynomial systems via online learning. Luo *et al.* (2015c) proposed an adaptive optimal control approach based on neuro-dynamic programming for the optimal control of general highly dissipative spatially distributed processes. However, the main drawback of dynamic programming methods today is the high computational complexity, which was named the 'curse of dimensionality' by Bellman (1957). It is well known that the continuous-time nonlinear optimal control problem depends on the solution to a typical Hamilton nonlinear partial differential equation (PDE), which is called the Hamilton-Jacobi-Bellman (HJB) equation. However, global analytic solutions to the HJB equation are difficult to obtain. A few methods have been proposed to obtain numerical solutions for the nonlinear optimal control problem. The reader can refer to, for example, Markman and Katz (2000), Sakamoto and van der Schaft (2008), Cacace *et al.* (2012), Wu and Luo (2012), Aguilar and Krener (2014), Govindarajan *et al.* (2014), Luo *et al.* (2014; 2015a; 2015b), Smears and Süli (2014), and the references therein. However, few of these algorithms can be applied to nonlinear CT systems in multi-dimensional spaces. What is more, the 'curse of dimensionality' will appear in most of these methods when they find

* Project supported by the China Postdoctoral Science Foundation (No. 2013M531842), the Natural Science Foundation of Guangdong Province, China (No. 2015A030313497), and the Science and Technology Program of Guangzhou, China (No. 2014KP000039)

ORCID: Xue-song CHEN, http://orcid.org/0000-0001-9530-0644

optimal numerical solutions satisfying a given accuracy in a required domain to the HJB equations.

In this paper, we propose a Galerkin approximation with Legendre polynomial (GALP) method for the generalized Hamilton-Jacobi-Bellman (GHJB) equation. The main advantage of the GALP algorithm is that much fewer integrals need to be computed and the feedback control laws can be implemented in different ways. The GAPL algorithm essentially computes the coefficients offline using Legendre base functions. Once the coefficients are calculated, the control laws will be obtained online. The idea of GAPL for the GHJB equation comes from Beard *et al.* (1997). The reader is also referred to Saridis and Lee (1979), Gong *et al.* (2006), Aguilar and Krener (2014), and Smears and Süli (2014) for details and different perspectives. An increasingly large quantity of computer memory is needed to store the coefficients in order to approximate the optimal solution when the number of base functions increases. Thus, how to choose the base functions for approximating the GHJB equation is essential for the optimal control problem. To the best of our knowledge, there is no literature on using the GALP algorithm for the GHJB equation. In our proposed computational method, the solution at each iteration using the Galerkin-Legendre spectral method can be approximated as an optimal control. Different from many existing Galerkin methods for solving PDE, this algorithm is not based on an odd symmetric function or finite element basis. A significant advantage of this approach is that much fewer integrals need to be computed because of the orthogonal properties of Legendre polynomials. The framework of the algorithm may be developed to overcome the effect of the 'curse of dimensionality' to some degree.

2 Problem statement

Consider the following CT affine nonlinear system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}, \tag{1}$$

where $\boldsymbol{x} \in \Omega \subset \mathbb{R}^n$ is the system state, $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$, $\boldsymbol{x}(\boldsymbol{0}) = \boldsymbol{x_0} \in \Omega \subset \mathbb{R}^n$ is the initial state and $\boldsymbol{u} \in \mathbb{R}^m$ is the control input, and $\boldsymbol{f} : \Omega \to \mathbb{R}^n$, $\boldsymbol{g} : \Omega \to \mathbb{R}^{n\times m}$, $\boldsymbol{u} : \Omega \to \mathbb{R}^m$. Suppose that the system $\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}$ is Lipschitz continuous on a set Ω, which is asymptotically stable for any initial condition in Ω.

The main objective of this study is to find the feedback control laws $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}^*(\boldsymbol{x})$ that satisfy Eq. (1), and to minimize a cost function as follows:

$$J(\boldsymbol{x_0}, \boldsymbol{u}) = \int_0^\infty \left[l(\boldsymbol{x}(t, \boldsymbol{x_0})) + ||\boldsymbol{u}(\boldsymbol{x}(t, \boldsymbol{x_0}))||^2_{\boldsymbol{R}}\right] \mathrm{d}t, \tag{2}$$

where $l : \Omega \to \mathbb{R}$ is the state penalty function, $\boldsymbol{R} \in \mathbb{R}^{m\times m}$ is a symmetric, positive-definite matrix, and $||\boldsymbol{u}||^2_{\boldsymbol{R}} = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}$ is the control penalty function. Given any admissible control $\boldsymbol{u}$, the value function at $\boldsymbol{x} \in \Omega$ is obtained as follows:

$$V(\boldsymbol{x}) = \int_0^\infty \left[l(\boldsymbol{x}(t, \boldsymbol{x_0})) + ||\boldsymbol{u}(\boldsymbol{x}(t, \boldsymbol{x_0}))||^2_{\boldsymbol{R}}\right] \mathrm{d}t. \tag{3}$$

Then we have the optimal control

$$\boldsymbol{u}^*(\boldsymbol{x}) = \arg\min_{\boldsymbol{u}} \ V(\boldsymbol{x}). \tag{4}$$

From the optimal control theory in Lews and Syrmos (1995) and Kirk (2012), it is easy to obtain the HJB equation for the optimal control problem:

$$\left(\frac{\partial V^*}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} \boldsymbol{f} + l - \frac{1}{4}\left(\frac{\partial V^*}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} \boldsymbol{g}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\frac{\partial V^*}{\partial \boldsymbol{x}} = 0, \tag{5}$$

with the boundary condition $V^*(\boldsymbol{0}) = \boldsymbol{0}$. Then we can design the optimal controller as follows:

$$\boldsymbol{u}^*(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\frac{\partial V^*}{\partial \boldsymbol{x}}. \tag{6}$$

It is well known that the optimal control function (6) depends on the solution V^* of Eq. (5). If system (1) is linear and the state penalty function is given as $l(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}$ with $\boldsymbol{Q} = \boldsymbol{Q}^{\mathrm{T}} \succeq 0$, then the optimal control problem becomes the linear quadratic regulator problem, and the solution is given by $V^* = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}$, where $\boldsymbol{P} = \boldsymbol{P}^{\mathrm{T}}$ satisfies the standard Riccati equation.

Although the solution to the nonlinear optimal problem has been well known since the early 1960s (Kleinman, 1968), relatively few control designs explicitly use a feedback function of the form given in Eq. (6). The primary difficulty lies in solving the HJB equation, for which general closed-form solutions do not exist. It is impossible to solve the HJB equations analytically. To obtain an approximate solution, Beard *et al.* (1997) proposed a Galerkin method to approximate the solution to the following GHJB equation:

$$\left(\frac{\partial V^{(i)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} (\boldsymbol{f} + \boldsymbol{g}\boldsymbol{u}^{(i)}) + l + ||\boldsymbol{u}^{(i)}||^2_{\boldsymbol{R}} = 0, \tag{7}$$

with

$$\boldsymbol{u}^{(i+1)}(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\frac{\partial V^{(i)}}{\partial \boldsymbol{x}}. \tag{8}$$

Given any initial control policy $\boldsymbol{u}^0$, the solution to Eq. (7) will converge to the solution to Eq. (5), which has been proven in Beard *et al.* (1997), i.e.,

$$\lim_{i\to\infty} V^{(i)} = V^*, \quad \lim_{i\to\infty} \boldsymbol{u}^{(i)} = \boldsymbol{u}^*. \tag{9}$$

Thus, the problem of finding the solution to the nonlinear HJB equation has been transformed into the problem of finding the solution to the linear GHJB equation. It is obvious that the GHJB equation is much easier to solve than the HJB equation. The Galerkin numerical approximation with a series of odd-symmetric bases was used to approximate the near-optimal solution to the GHJB equation in Beard *et al.* (1997). The main disadvantage of this method is that $O(N^2)$ n-dimensional integrals need to be calculated, which is inherently complex. In addition, for a given ϵ, the large number N cannot be estimated in this approach. The main contribution of this study is that, for a well-defined problem of optimal control, it is easier to achieve optimal feedback control laws using the GALP algorithm, and this approach requires much fewer integrals to be computed because of the orthogonal properties.

3 Galerkin approximation with Legendre polynomial method

In this section, the GHJB equation will be solved using the Galerkin approximation technique proposed by Saridis and Lee (1979) and Beard *et al.* (1997). There is a lot of literature on solutions to HJB equations, including the classic viscosity solutions in Bardi and Capuzzo-Dolcetta (1997). Note that the GHJB equation may have several continuous solutions. However, only one of these solutions is positive definite and produces a stable control by Eq. (4). The successive approximation scheme with a series of basis functions is optimal with respect to the cost index specified. Also, note that if the algorithm is truncated for any $i < \infty$, $V^{(i)}$ will be a kind of Lyapunov function for system (1). This means that the algorithm can be stopped at any point but still results in a stable control that has better performance than $\boldsymbol{u}^{(0)}$. Then we will derive the GALP method to solve the GHJB equation associated with the optimal control problem.

3.1 Algorithm procedure based on Legendre polynomial

Assume that there exists a set $\Phi = \{\phi_j\}_{j=1}^{\infty}$ where $\phi_j : \Omega \to \mathbb{R}$ and $\phi_j(0) = 0$, which are not necessarily linearly independent and satisfy the boundary condition. It implies that every solution to the GHJB equation can be represented as

$$V^{(i)}(\boldsymbol{x}) = \sum_{j=1}^{\infty} c_j^{(i)}\phi_j(\boldsymbol{x}). \tag{10}$$

A finite truncation of this sum represents an approximation of $V(\boldsymbol{x})$. The required degree of accuracy in the approximation can be made by selecting a large enough number N as follows:

$$V_N^{(i)}(\boldsymbol{x}) = \sum_{j=1}^{N} c_j^{(i)}\phi_j(\boldsymbol{x}). \tag{11}$$

It is very important to select the basis function for system (1). Generally speaking, if an infinite series of basis functions were used, any set would suffice. However, the truncation of the infinite series results in approximation errors and elevates the importance of carefully selecting a set of equations that closely represent the true solution. Here we use the GALP approach to solve the GHJB equation.

We suppose that a complete set of Legendre base functions $\{\phi_j\}_{j=1}^{\infty}$ which satisfy $V(\boldsymbol{x}) = \sum_{j=1}^{\infty} c_j\phi_j(\boldsymbol{x})$ can be found. A near-optimal V is given by truncating the series to $V_N(\boldsymbol{x}) = \sum_{j=1}^{N} c_j\phi_j(\boldsymbol{x})$. The approximation with a series of Legendre bases is valid when the base functions are defined on $[-1, 1]$. If the system is defined on the interval $[a, b]$, we can transform it to the interval $[-1, 1]$ by using linear transformation. We will find a numerical solution which is the polynomial with a degree N. In fact, if N is sufficiently large, then $e_N(\boldsymbol{x}) \to 0$ and thus the near-optimal solution can be obtained as a convergent Legendre polynomial series.

It is well known that a set of Legendre base functions $\{\phi_0, \phi_1, \cdots, \phi_N\}$ are orthogonal on $[-1, 1]$ with weight function $\Omega(\boldsymbol{x}) = 1$. Legendre base functions ϕ_N satisfy the following equation:

$$\phi_{n+1}'(\boldsymbol{x}) - \phi_n'(\boldsymbol{x}) = (2n+1)\phi_n(\boldsymbol{x}),\ n \geq 1, \tag{12}$$

where

$$[n/2] = \begin{cases} \frac{n}{2}, & n \text{ is even}, \\ \frac{n-1}{2}, & n \text{ is odd}, \end{cases} \tag{13}$$

or

$$\phi_n(\boldsymbol{x}) = \frac{1}{2^n}\sum_{k=1}^{[n/2]}(-1)^k \begin{bmatrix} n \\ k \end{bmatrix} \begin{bmatrix} 2n-2k \\ n \end{bmatrix} \boldsymbol{x}^{N-2k}, \quad (14)$$

where $n = 0, 1, \cdots$. In addition, they satisfy the following orthogonal properties ($i, j = 0, 1, \cdots$):

$$\int_{-1}^{1} \phi_i(s)\phi_j(s)\mathrm{d}s = \begin{cases} 0, & i \neq j, \\ \frac{2}{2n+1}, & i = j. \end{cases} \quad (15)$$

Next we will show the numerical solutions to the GHJB equation obtained by using the GALP algorithm. Assuming the infinite sum in Eq. (10) satisfies the GHJB equation, we have

$$\left(\sum_{j=1}^{\infty} c_j^{(i)} \left(\frac{\partial \phi_j}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\right)(\boldsymbol{f}+\boldsymbol{g}\boldsymbol{u}^{(i)}) + l + ||\boldsymbol{u}^{(i)}||_{\boldsymbol{R}}^2 = 0. \quad (16)$$

Accordingly, the finite truncation of $V^{(i)}(\boldsymbol{x})$ satisfies

$$\left(\sum_{j=1}^{N} c_j^{(i)} \left(\frac{\partial \phi_j}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\right)(\boldsymbol{f}+\boldsymbol{g}\boldsymbol{u}^{(i)}) + l + ||\boldsymbol{u}^{(i)}||_{\boldsymbol{R}}^2 = e_N^{(i)}(\boldsymbol{x}), \quad (17)$$

where $e_N^{(i)}(\boldsymbol{x})$ is the error introduced by truncation. The GALP scheme is to find coefficients c_j ($j = 1, 2, \cdots, N$) such that the projection of error $e_N^{(i)}(\boldsymbol{x})$ onto the finite linear space $\mathrm{span}\{\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \cdots, \phi_N(\boldsymbol{x})\}$ is zero for any $\boldsymbol{x} \in \Omega$.

Therefore, the coefficients $c_1, c_2, \cdots, c_N$ can be found from the following set of algebraic equations:

$$\langle e_N^{(i)}(\boldsymbol{x}), \phi_j(\boldsymbol{x})\rangle = 0, \ j = 1, 2, \cdots, N, \quad (18)$$

where the function inner product is

$$\langle \boldsymbol{f}, \boldsymbol{g}\rangle = \int_{\Omega} \boldsymbol{f}(s)\boldsymbol{g}(s)\mathrm{d}s. \quad (19)$$

Eq. (18) can also be written as

$$\begin{bmatrix} \eta_{11}^{(i)} & \cdots & \eta_{1N}^{(i)} \\ \vdots & & \vdots \\ \eta_{N1}^{(i)} & \cdots & \eta_{NN}^{(i)} \end{bmatrix} \begin{bmatrix} c_1^{(i)} \\ \vdots \\ c_N^{(i)} \end{bmatrix} = -\begin{bmatrix} \langle l + ||\boldsymbol{u}^{(i)}||_{\boldsymbol{R}}^2, \phi_1\rangle \\ \vdots \\ \langle l + ||\boldsymbol{u}^{(i)}||_{\boldsymbol{R}}^2, \phi_N\rangle \end{bmatrix}, \quad (20)$$

where

$$\eta_{jk}^{(i)} = \left\langle \left(\frac{\partial \phi_j}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} (\boldsymbol{f}+\boldsymbol{g}\boldsymbol{u}^{(i)}), \phi_k \right\rangle. \quad (21)$$

If we define matrices $\boldsymbol{M}_1$ and $\boldsymbol{M}_2^{(i)}$ as follows:

$$\boldsymbol{M}_1 = \begin{bmatrix} \left\langle \left(\frac{\partial \phi_1}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} \boldsymbol{f}, \phi_1 \right\rangle & \cdots & \left\langle \left(\frac{\partial \phi_N}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} \boldsymbol{f}, \phi_1 \right\rangle \\ \vdots & & \vdots \\ \left\langle \left(\frac{\partial \phi_1^{\mathrm{T}}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} \boldsymbol{f}, \phi_N \right\rangle & \cdots & \left\langle \left(\frac{\partial \phi_N^{\mathrm{T}}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} \boldsymbol{f}, \phi_N \right\rangle \end{bmatrix}, \quad (22)$$

$$\boldsymbol{M}_2^{(i)} = \begin{bmatrix} \left\langle \left(\frac{\partial \phi_1^{\mathrm{T}}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} \boldsymbol{g}\boldsymbol{u}^{(i)}, \phi_1 \right\rangle & \cdots & \left\langle \left(\frac{\partial \phi_N^{\mathrm{T}}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} \boldsymbol{g}\boldsymbol{u}^{(i)}, \phi_1 \right\rangle \\ \vdots & & \vdots \\ \left\langle \left(\frac{\partial \phi_1^{\mathrm{T}}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} \boldsymbol{g}\boldsymbol{u}^{(i)}, \phi_N \right\rangle & \cdots & \left\langle \left(\frac{\partial \phi_N^{\mathrm{T}}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}} \boldsymbol{g}\boldsymbol{u}^{(i)}, \phi_N \right\rangle \end{bmatrix}, \quad (23)$$

$$\boldsymbol{b}_1 = -\begin{bmatrix} \langle l, \phi_1\rangle \\ \vdots \\ \langle l, \phi_N\rangle \end{bmatrix}, \quad (24)$$

$$\boldsymbol{b}_2^{(i)} = -\begin{bmatrix} \langle ||\boldsymbol{u}^{(i)}||_{\boldsymbol{R}}^2, \phi_1\rangle \\ \vdots \\ \langle ||\boldsymbol{u}^{(i)}||_{\boldsymbol{R}}^2, \phi_N\rangle \end{bmatrix}, \quad (25)$$

then we can rewrite Eq. (20) as

$$(\boldsymbol{M}_1 + \boldsymbol{M}_2^{(i)})\boldsymbol{c}^{(i)} = \boldsymbol{b}_1 + \boldsymbol{b}_2^{(i)}. \quad (26)$$

Hence, a near-optimal approximation of the solution to Eq. (7) can be produced by computing $c_j^{(i)}$. The approximate control associated with Eq. (6) is given by

$$\boldsymbol{u}_N^{(i+1)} = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\left(\sum_{j=1}^{N} c_j^{(i)} \frac{\partial \phi_j}{\partial \boldsymbol{x}}\right). \quad (27)$$

Now the complete GALP algorithm procedure is proposed to solve the GHJB equation:

Step 1: Suppose $\boldsymbol{u}^{(0)}$ is an initial control for system (1). Input $\boldsymbol{f}, \boldsymbol{g}, l, \boldsymbol{R}, \Omega$ and the base functions ϕ_j.

Step 2: Compute the initial values of $\boldsymbol{M}_1$, $\boldsymbol{M}_2^{(0)}$, $\boldsymbol{b}_1$, $\boldsymbol{b}_2^{(0)}$.

Step 3: Compute $\boldsymbol{c}^{(i)}$ according to Eq. (26), compute the value function $V^{(i)}$ with Eq. (11), and update the control $\boldsymbol{u}^{(i)}$ with Eq. (27).

Step 4: Let $i = i + 1$. If $|V^{(i)} - V^{(i-1)}| \leq \epsilon$ (ϵ is a small enough positive number), then stop the iteration and employ $\boldsymbol{u}^{(i)}$ to obtain the final control law with Eq. (6); otherwise, go back to step 3.

Note that coefficients $\boldsymbol{c}^{(i)}$ can be computed offline prior to implementing the control.

3.2 Stability and convergence analysis

In this subsection, we analyze the convergence and stability of the proposed scheme. The Hamiltonian function is defined as follows:

$$\begin{aligned}&H(\boldsymbol{x},\frac{\partial V}{\partial \boldsymbol{x}},\boldsymbol{u},t)\\&=l(\boldsymbol{x})+||\boldsymbol{u}||^2_{\boldsymbol{R}}+\left(\frac{\partial V}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}(\boldsymbol{f}(\boldsymbol{x})+\boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}).\end{aligned}\tag{28}$$

Lemma 1 (Theorem 1 in Saridis and Lee (1979)) Assume $\boldsymbol{u}^{(1)}(t,\boldsymbol{x})\in\Omega_{\boldsymbol{u}}$, where $t\in[0,T]$ is an admissible and arbitrary feedback control. If the function $V(\boldsymbol{x};t,\boldsymbol{u}^{(1)})$ is positive definite and differentiable on $[0,T]\times\mathbb{R}^n$, which satisfies

$$\frac{\partial V}{\partial t}+H(\boldsymbol{x},\frac{\partial V}{\partial \boldsymbol{x}},\boldsymbol{u}^{(1)},t)=0,\tag{29}$$

$$V(\boldsymbol{x}(T);t,\boldsymbol{u}^{(1)})=s(\boldsymbol{x}(T)),\tag{30}$$

then the value function of system (1) is $V(\boldsymbol{x};t,\boldsymbol{u}^{(1)})$, and

$$V^{(1)}(\boldsymbol{x})=J(\boldsymbol{u}^{(1)})=\int_0^\infty[l(\boldsymbol{x})+||\boldsymbol{u}^{(1)}||^2_{\boldsymbol{R}}]\mathrm{d}t.\tag{31}$$

Lemma 2 If the optimal value function $V^*(\boldsymbol{x})$ and optimal control $\boldsymbol{u}^*$ exist, and satisfy Lemma 1, then

$$0<V^*(\boldsymbol{x})\le V(\boldsymbol{x}),\ \boldsymbol{u}\ne\boldsymbol{u}^*.\tag{32}$$

Proof The proof of Lemma 2 is easy to obtain from Lemma 1.

There is a large amount of literature devoted to designing stable regulators for nonlinear systems. The most important and popular tool is Lyapunov's method. To use Lyapunov's method, a designer first proposes a control and then tries to find a Lyapunov function for the closed-loop system. A Lyapunov function is a generalized energy function of the states, and is usually suggested by the physics of the problem. It is often possible to find a stable control for a particular system. Now we show that value function $V^{(1)}$ is a Lyapunov function of system $(\boldsymbol{f},\boldsymbol{g},\boldsymbol{u}^{(2)})$.

Theorem 1 Assume Ω is a compact set, and $\boldsymbol{u}^{(1)}\in A_{\boldsymbol{u}}(\Omega)$ is an arbitrary admissible control for system (1) on Ω, if a positive definite, differentiable function $V^{(1)}(t,\boldsymbol{x})$ exists on $[0,\infty)\times\mathbb{R}^n$ and satisfies the GHJB equation

$$\left(\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}(\boldsymbol{f}+\boldsymbol{g}\boldsymbol{u}^{(1)})+l+||\boldsymbol{u}^{(1)}||^2_{\boldsymbol{R}}=0,\tag{33}$$

associated with

$$\boldsymbol{u}^{(2)}=-\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\frac{\partial V^{(1)}}{\partial \boldsymbol{x}},\tag{34}$$

then $\forall\, t\in[0,\infty)$, system (1) has bounded trajectories over $[0,\infty)$, and the stable equilibrium point is the origin in system (1).

Proof Since $V^{(1)}(t,\boldsymbol{x})$ is a continuously differentiable function, we take the derivative of $V^{(1)}(t,\boldsymbol{x})$ along system $(\boldsymbol{f},\boldsymbol{g},\boldsymbol{u}^{(2)})$ and follow the equation $\left(\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\boldsymbol{g}=-2(\boldsymbol{u}^{(2)})^{\mathrm{T}}\boldsymbol{R}$. Then we have

$$\begin{aligned}\dot{V}_1(t,\boldsymbol{x})&=\frac{\partial V^{(1)}}{\partial t}+\left(\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\dot{\boldsymbol{x}}\\&=\frac{\partial V^{(1)}}{\partial t}+\left(\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}(\boldsymbol{f}+\boldsymbol{g}\boldsymbol{u}^{(2)})\\&=\frac{\partial V^{(1)}}{\partial t}+\left(\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\boldsymbol{f}+\left(\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\boldsymbol{g}\boldsymbol{u}^{(2)}\\&=\frac{\partial V^{(1)}}{\partial t}+\left(\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\boldsymbol{f}-2(\boldsymbol{u}^{(2)})^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}^{(2)}\\&=\frac{\partial V^{(1)}}{\partial t}+\left(\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\boldsymbol{f}-2||\boldsymbol{u}^{(2)}||^2_{\boldsymbol{R}}.\end{aligned}\tag{35}$$

When $t\to\infty$, $\frac{\partial V^{(1)}}{\partial t}=0$, the following equation is obtained from the GHJB equation:

$$\left(\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\boldsymbol{f}=-l-||\boldsymbol{u}^{(1)}||^2_{\boldsymbol{R}}-\left(\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\boldsymbol{g}\boldsymbol{u}^{(1)}.\tag{36}$$

Substituting Eq. (36) into Eq. (35), we obtain

$$\begin{aligned}&\dot{V}_1(t,\boldsymbol{x})\\&=-l-||\boldsymbol{u}^{(1)}||^2_{\boldsymbol{R}}-\left(\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}\right)^{\mathrm{T}}\boldsymbol{g}\boldsymbol{u}^{(1)}-2||\boldsymbol{u}^{(2)}||^2_{\boldsymbol{R}}\\&=-l-||\boldsymbol{u}^{(1)}||^2_{\boldsymbol{R}}+2(\boldsymbol{u}^{(2)})^{\mathrm{T}}\boldsymbol{g}\boldsymbol{u}^{(1)}-2||\boldsymbol{u}^{(2)}||^2_{\boldsymbol{R}}\\&=-l-\boldsymbol{u}_2^{\mathrm{T}}\boldsymbol{R}\boldsymbol{u}^{(2)}-(\boldsymbol{u}^{(1)}-\boldsymbol{u}^{(2)})^{\mathrm{T}}\boldsymbol{R}(\boldsymbol{u}^{(1)}-\boldsymbol{u}^{(2)})\\&=-l-||\boldsymbol{u}^{(2)}||^2_{\boldsymbol{R}}-||\boldsymbol{u}^{(1)}-\boldsymbol{u}^{(2)}||^2_{\boldsymbol{R}}\\&<0.\end{aligned}\tag{37}$$

The boundaries of the trajectories are established according to system (1) over $[0,\infty)$. Under the assumptions of the theorem, $\dot{V}^{(1)}(t,\boldsymbol{x})<0$ is a Lyapunov function, and the stable equilibrium point is the origin of system (1).

We will show that if we choose an updated control according to our proposed algorithm, then the new control improves the performance and is admissible. Lemma 3 in Beard *et al.* (1997) is a strengthened version of the theorem in Saridis and Lee (1979) of which the associated $\boldsymbol{u}^{(2)}$ has been selected to satisfy

$$||\boldsymbol{u}^{(2)} + \frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\frac{\partial V^{(2)}}{\partial \boldsymbol{x}}|| \leq ||\boldsymbol{u}^{(1)} + \frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\frac{\partial V^{(1)}}{\partial \boldsymbol{x}}||. \tag{38}$$

However, it is not necessary to solve the GHJB equation by our method to satisfy the above inequality. To obtain the construction of a feasible solution developed in the following Theorem 2, we need the following lemma in Gong *et al.* (2006):

Lemma 3 Given an arbitrary function $\xi(t) \in W^{m,\infty}$, for any $t \in [-1,1]$, there is a polynomial $p^N(t)$ with degree N which satisfies

$$|\xi(t) - p^N(t)| \leq C_0 C N^{-m},\ \forall\, t \in [-1,1], \tag{39}$$

where $C_0 = ||\xi||_{W^{m,\infty}}$ and C is a constant independent of N. $p^N(t)$ is called the best Nth-order polynomial approximation of $\xi(t)$ in the L_∞ norm.

Proof It is a standard result of polynomial approximations (Canuto *et al.*, 1988).

In the GALP method, the main idea is to approximate $V(\boldsymbol{x}(t))$ by the Nth-order Legendre polynomial $\phi_N(\boldsymbol{x}(t))$. Let $t_0 = -1 < t_1 < \cdots < t_N = 1$ be the nodes defined as t_k which are roots of $\dot{L}_N(t)$, $k = 1,2,\cdots,N-1$, where $\dot{L}_N(t)$ is the derivative of the Nth-order Legendre polynomial $L_N(t)$. Let $\overline{V}_k^N$ and $\overline{\boldsymbol{u}}_k^N$ be an approximation of a feasible solution $V(\boldsymbol{x}(t))$ evaluated at node t_k. Then we can obtain the following theorem:

Theorem 2 Given an arbitrary feasible solution $\boldsymbol{x} \to (V^{(i)}, \boldsymbol{u}^{(i)})$ for Eq. (7), assume that $\boldsymbol{x}(t) \in W^{m,\infty}$ with $m \geq 2$, and there exists an integer N_0 such that $N > N_0$ for any N. Then the GHJB equation has a feasible solution $(\overline{V}_k^N, \overline{\boldsymbol{u}}_k^N)$, $k = 0,1,\cdots,N$, which satisfies

$$|V(t_k) - \overline{V}_k^N| \leq L(N-n)^{1-m}, \tag{40}$$

$$|\boldsymbol{u}(t_k) - \overline{\boldsymbol{u}}_k^N| \leq L(N-n)^{1-m}, \tag{41}$$

$\forall\, k = 0,1,\cdots,N$, where L is a constant.

Proof Assume that $L(\boldsymbol{x}(t))$ is the $(N-n)$th-order optimal approximation of $\dot{V}^{(i)}(\boldsymbol{x}(t))$ with respect to the norm of L_∞. According to Lemma 3, there exists a constant C_0 independent of N such that

$$|\dot{V}^{(i)}(\boldsymbol{x}(t)) - L(\boldsymbol{x}(t))| \leq C_0(N-n)^{1-m},\ \forall\, t \in [-1,1]. \tag{42}$$

Define

$$\hat{V}^{(n)}(\boldsymbol{x}(t)) = \int_{-1}^{T} L(\boldsymbol{x}(\tau))\mathrm{d}\tau + V^{(n)}(-1), \tag{43}$$

$$\hat{V}^{(n-1)}(\boldsymbol{x}(t)) = \int_{-1}^{T} \hat{V}^{(n)}(\tau)\mathrm{d}\tau + V^{(n-1)}(-1), \tag{44}$$

$$\vdots$$

$$\hat{V}^{(1)}(\boldsymbol{x}(t)) = \int_{-1}^{T} \hat{V}^{(2)}(\tau)\mathrm{d}\tau + V^{(1)}(-1), \tag{45}$$

$$\hat{\boldsymbol{u}}(\boldsymbol{x}(t)) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\dot{V}^{(i)}(\boldsymbol{x}(t)). \tag{46}$$

Clearly, $\hat{V}^{(1)}(\boldsymbol{x}(t)), \hat{V}^{(2)}(\boldsymbol{x}(t)), \cdots, \hat{V}^{(n)}(\boldsymbol{x}(t))$ are polynomials of degrees equal to or less than N, which satisfy the differential Eq. (1). By definition, the derivatives of a polynomial of degree N evaluated at nodes $t_0, t_1, \cdots, t_N$ are equal to the values of the polynomials at the nodes multiplied by the differentiation matrix $\boldsymbol{D}$ in Canuto *et al.* (1988). Thus, if we let

$$\overline{V}_k^N = \hat{V}(t_k),\ \overline{\boldsymbol{u}}_k^N = \hat{\boldsymbol{u}}(t_k), \tag{47}$$

we have

$$\boldsymbol{D}\begin{bmatrix}\bar{\boldsymbol{x}}_{i,0}^N\\ \vdots\\ \bar{\boldsymbol{x}}_{i,N}^N\end{bmatrix} = \boldsymbol{D}\begin{bmatrix}\hat{\boldsymbol{x}}_i(t_0)\\ \vdots\\ \hat{\boldsymbol{x}}_i(t_N)\end{bmatrix} = \begin{bmatrix}\dot{\hat{\boldsymbol{x}}}_i(t_0)\\ \vdots\\ \dot{\hat{\boldsymbol{x}}}_i(t_N)\end{bmatrix} = \begin{bmatrix}\hat{\boldsymbol{x}}_{i+1}(t_0)\\ \vdots\\ \hat{\boldsymbol{x}}_{i+1}(t_N)\end{bmatrix} = \begin{bmatrix}\bar{\boldsymbol{x}}_{i+1,0}^N\\ \vdots\\ \bar{\boldsymbol{x}}_{i+1,N}^N\end{bmatrix}, \tag{48}$$

where $i = 1,2,\cdots,n-1$ and $\bar{\boldsymbol{x}}_{i,k}^N$ is the ith component of $\bar{\boldsymbol{x}}_k^N$. At $i = n$, we have

$$\boldsymbol{D}\begin{bmatrix}\bar{\boldsymbol{x}}_{n,0}^N\\ \vdots\\ \bar{\boldsymbol{x}}_{n,N}^N\end{bmatrix} = \boldsymbol{D}\begin{bmatrix}\hat{\boldsymbol{x}}_n(t_0)\\ \vdots\\ \hat{\boldsymbol{x}}_n(t_N)\end{bmatrix} = \begin{bmatrix}\dot{\hat{\boldsymbol{x}}}_n(t_0)\\ \vdots\\ \dot{\hat{\boldsymbol{x}}}_n(t_N)\end{bmatrix} = \begin{bmatrix}\boldsymbol{f}(\hat{\boldsymbol{x}}(t_0)) + \boldsymbol{g}(\hat{\boldsymbol{x}}(t_0))\hat{\boldsymbol{u}}(t_0)\\ \vdots\\ \boldsymbol{f}(\hat{\boldsymbol{x}}(t_N)) + \boldsymbol{g}(\hat{\boldsymbol{x}}(t_0))\hat{\boldsymbol{u}}(t_N)\end{bmatrix}. \tag{49}$$

Therefore, $(\bar{\boldsymbol{x}}_k^N, \bar{\boldsymbol{u}}_k^N)$, $k = 0,1,\cdots,N$, satisfy system (1). Next, we prove that the above inequalities are also satisfied. We can obtain the following inequalities based on Gong *et al.* (2006):

$$\begin{cases} |V^{(n)}(\boldsymbol{x}(t)) - \hat{V}^{(n)}(\boldsymbol{x}(t))| \leq 2C_0(N-n)^{1-m}, \\ \qquad \vdots \\ |V^{(1)}(\boldsymbol{x}(t)) - \hat{V}^{(1)}(\boldsymbol{x}(t)| \leq 2^n C_0(N-n)^{1-m}. \end{cases} \tag{50}$$

Supposing $m \geq 2$, both $V(\boldsymbol{x}(t))$ and $\hat{V}(\boldsymbol{x}(t))$ are contained in a kind of compact set. Furthermore, as $\boldsymbol{f}$ and $\boldsymbol{g}$ are Lipschitz continuous in the compact set, there exists a constant C_1 independent of N such that

$$\begin{aligned} &|\boldsymbol{u}(\boldsymbol{x}(t)) - \hat{\boldsymbol{u}}(\boldsymbol{x}(t))| \\ &= \frac{1}{2}|\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\dot{\hat{V}}(\boldsymbol{x}(t)) - \boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}\dot{V}(\boldsymbol{x}(t))| \\ &\leq C_1\big(|\dot{V}^{(n)}(\boldsymbol{x}(t)) - p(t)| + |V^{(1)}(\boldsymbol{x}(t)) - \hat{V}^{(1)}(\boldsymbol{x}(t))| \\ &\quad + \cdots + |V^{(n)}(\boldsymbol{x}(t)) - \hat{V}^{(n)}(\boldsymbol{x}(t))|\big). \end{aligned} \tag{51}$$

Hence, we have the following inequalities:

$$|V^{(i)}(\boldsymbol{x}(t)) - \hat{V}(\boldsymbol{x}(t))| \leq C_2(N-n)^{1-m}, \tag{52}$$

$$|\boldsymbol{u}(\boldsymbol{x}(t)) - \hat{\boldsymbol{u}}(\boldsymbol{x}(t))| \leq C_2(N-n)^{1-m}, \tag{53}$$

$\forall\, i = 1, 2, \cdots, n$, $t \in [-1, 1]$ and for some positive constant C_2 independent of N. By a similar procedure, it is proved that the endpoint condition is satisfied. Hence, $(V^{(i)}, \boldsymbol{u}^{(i)})$ is a feasible discredited solution. At $t = t_k$, Eqs. (52) and (53) imply Eqs. (40) and (41). Thus, we have constructed a feasible solution to system (1) that satisfies Eqs. (40) and (41).

4 Illustrative examples

Illustrative examples are aimed to demonstrate the effectiveness and usefulness of the proposed method for designing controllers. To do this, we will show how the GALP algorithm is used to solve the GHJB equation and obtain the optimal control laws for nonlinear CT systems.

4.1 Comparison with the odd-symmetric basis method

In this subsection we will compare the control law obtained using our GALP algorithm with the odd-symmetric basis method proposed by Beard *et al.* (1996). The system is considered as follows:

$$\begin{cases} \dot{x}_1(t) = x_2(t), \\ \dot{x}_2(t) = \epsilon\, x_1^3(t) + u(t), \end{cases} \tag{54}$$

subject to cost function

$$J = \frac{1}{2}\int_0^\infty (x_1^2 + x_2^2 + u^2)\mathrm{d}t. \tag{55}$$

Beard *et al.* (1996) solved this problem using a Galerkin approximation method with an odd-symmetric basis. When $\epsilon = 1$, the GHJB equation for system (54) is

$$H(\boldsymbol{x}, \boldsymbol{p}) = x_2 p_1 + x_1^3 p_2 - \frac{1}{2}p_2^2 + \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 = 0, \tag{56}$$

where $\boldsymbol{x} = [x_1, x_2]^{\mathrm{T}}$, $\boldsymbol{p} = \frac{\partial V}{\partial \boldsymbol{x}} = [p_1, p_2]^{\mathrm{T}}$. The optimal control for system (54) is easy to obtain as $u^* = p_2$. Beard *et al.* (1996) developed fourth-order approximations for this system:

$$\begin{aligned} u_{\mathrm{B}}(\boldsymbol{x}) = &-1.0376x_1 - 1.7975x_2 - 1.3079x_1^3 \\ &- 1.3429x_1^2x_2 - 0.4664x_1x_2^2 - 0.74x_2^3. \end{aligned} \tag{57}$$

The control is compared to the solution obtained by the GALP approach for $x_1(0) \in (-1, 1)$ and $x_2(0) = 0$. We use Legendre polynomials up to degree 4 as basis functions, and when GALP is applied to the system, the following control is calculated for $i = 9$:

$$\begin{aligned} u_{\mathrm{G}}(\boldsymbol{x}) = &-1.0105x_1 - 1.6326x_2 - 0.6245x_1^3 \\ &- 1.5325x_1^2x_2 - 0.8513x_1x_2^2 - 0.3136x_2^3. \end{aligned} \tag{58}$$

Figs. 1 and 2 show the results obtained using Beard's control method and the GALP control method, respectively. In these figures, u_{B} refers to Beard's method, and u_{G} refers to our proposed control method. The optimal feedback control obtained by the GALP method is better than that obtained by Beard *et al.* (1996)'s method. Moreover, the computational cost of the approximation optimal control with the GALP approach can be reduced when the stop criterion is strengthened.

4.2 Comparison with the exact linearization method

In this subsection we will compare our method with the method of exact feedback linearization. We use the following system:

$$\begin{cases} \dot{x}_1(t) = -x_1^3(t) - x_2(t), \\ \dot{x}_2(t) = x_1(t) + x_2(t) + u(t). \end{cases} \tag{59}$$

The control objective is to minimize the quadratic cost function

$$J = \int_0^\infty \left[\boldsymbol{x}^{\mathrm{T}}(t)\boldsymbol{x}(t) + u^2(\boldsymbol{x}(t))\right] \mathrm{d}t. \tag{60}$$

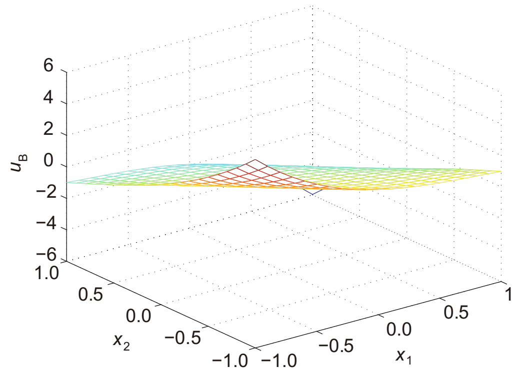

Fig. 1 u_B for the optimal control problem obtained using Beard's control method

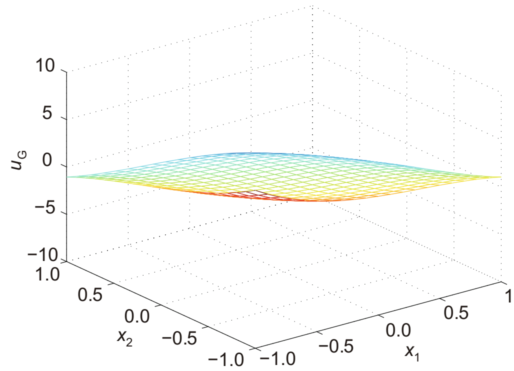

Fig. 2 u_G for the optimal control problem obtained using the GALP control method

Isidori (2013) linearized this system using the state feedback as follows:

$$u(\boldsymbol{x}) = 3x_1^5 + 3x_1^2x_2 - x_2 + v \tag{61}$$

and the coordinate transformation

$$\begin{cases} z_1 = -x_1, \\ z_2 = x_1^3 + x_2. \end{cases} \tag{62}$$

In the new coordinates, system (59) becomes

$$\begin{cases} \dot{z}_1(t) = -z_2(t), \\ \dot{z}_2(t) = z_1(t) + v. \end{cases} \tag{63}$$

Now we can obtain the optimal control for the transformed system using LQR theory with respect to the cost function

$$J = \int_0^\infty \left[\boldsymbol{z}^\mathrm{T}(t)\boldsymbol{z}(t) + v^2(\boldsymbol{z}(t))\right] \mathrm{d}t. \tag{64}$$

It is easy to obtain the control law

$$v(\boldsymbol{z}) = 0.4142z_1 - 1.3522z_2. \tag{65}$$

The suboptimal control synthesized by exact linearization is given by substituting Eq. (62) into Eq. (65) to obtain $v(\boldsymbol{x})$, and then using Eq. (61) to obtain

$$u_\mathrm{I}(\boldsymbol{x}) = 3x_1^5 + 3x_1^2x_2 - x_2 + 0.4142x_1 - 1.3522x_1^3 - 1.3522x_2. \tag{66}$$

The control $u_\mathrm{I}(\boldsymbol{x})$ is stable on $\mathbb{R}^2$. Because Ω is a compact set, we can let $\Omega = [-1,1] \times [-1,1]$ for simplicity. In this example we use Legendre base functions to approximate the solution to the GHJB equation instead of the odd-symmetric basis functions in Beard *et al.* (1997). Using our method to a sixth-order approximation, we obtain the following feedback control:

$$\begin{aligned} u_\mathrm{G}^{(6)}(\boldsymbol{x}) &= 0.1753x_1 - 2.6765x_2 - 0.7588x_1x_2^2 + 0.5018x_2^3 \\ &\quad - 1.5638x_1^2x_2 - 0.8645x_1^3 + 0.7913x_1x_2^4 \\ &\quad - 0.0628x_2^5 - 0.2687x_1^2x_2^3 + 0.5372x_1^3x_2^2 \\ &\quad - 0.7635x_1^4x_2 + 0.6820x_1^5. \end{aligned} \tag{67}$$

The value function V_I with respect to control u_I is compared with the cost function $V_\mathrm{G}^{(6)}$ associated with control $u_\mathrm{G}^{(6)}$ obtained from the GALP algorithm (Fig. 3). In this example, it is shown that our method performs much better than exact feedback linearization.

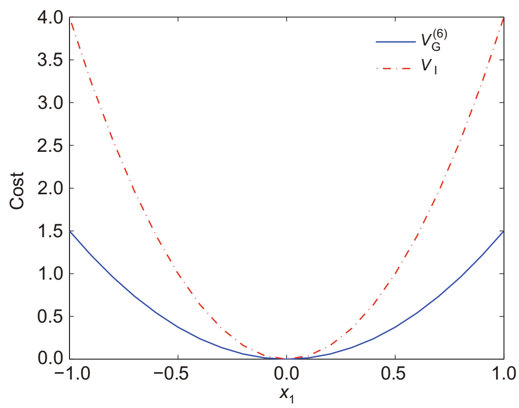

Fig. 3 Comparison with feedback linearization

5 Conclusions

It is important to have a numerical algorithm to approximate the solution to the HJB equation in the nonlinear optimal control problem. A new GALP method is proposed which provides a suboptimal solution to the GHJB equation. The method can be used to find a numerical solution to the GHJB equation. The resulting control laws are stable and converge to the optimal control laws.

References

Aguilar, C.O., Krener, A.J., 2014. Numerical solutions to the Bellman equation of optimal control. *J. Optim. Theory Appl.*, **160**(2):527-552. https://doi.org/10.1007/s10957-013-0403-8

Bardi, M., Capuzzo-Dolcetta, I., 1997. Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations. Birkhäuser Boston, Inc., Boston, MA (with appendices by Maurizio Falcone and Pierpaolo Soravia). https://doi.org/10.1007/978-0-8176-4755-1

Beard, R.W., Saridis, G.N., Wen, J.T., 1996. Improving the performance of stabilizing controls for nonlinear systems. *IEEE Contr. Syst. Mag.*, **16**(5):27-35. https://doi.org/10.1109/37.537206

Beard, R.W., Saridis, G.N., Wen, J.T., 1997. Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation. *Automatica*, **33**(12):2159-2177. https://doi.org/10.1016/S0005-1098(97)00128-3

Bellman, R., 1957. Dynamic Programming. Princeton University Press, New Jersey, USA.

Cacace, S., Cristiani, E., Falcone, M., *et al.*, 2012. A patchy dynamic programming scheme for a class of Hamilton-Jacobi-Bellman equations. *SIAM J. Sci. Comput.*, **34**(5):A2625-A2649. https://doi.org/10.1137/110841576

Canuto, C., Hussaini, M.Y., Quarteroni, A., *et al.*, 1988. Spectral Methods in Fluid Dynamics. Springer-Verlag, New York, USA.

Gong, Q., Kang, W., Ross, I.M., 2006. A pseudospectral method for the optimal control of constrained feedback linearizable systems. *IEEE Trans. Autom. Contr.*, **51**(7):1115-1129. https://doi.org/10.1109/TAC.2006.878570

Govindarajan, N., de Visser, C.C., Krishnakumar, K., 2014. A sparse collocation method for solving time-dependent HJB equations using multivariate B-splines. *Automatica*, **50**(9):2234-2244. https://doi.org/10.1016/j.automatica.2014.07.012

Isidori, A., 2013. Nonlinear Control Systems. Springer Science & Business Media.

Kirk, D.E., 2012. Optimal Control Theory: an Introduction. Courier Corporation.

Kleinman, D., 1968. On an iterative technique for Riccati equation computations. *IEEE Trans. Autom. Contr.*, **13**(1):114-115. https://doi.org/10.1109/TAC.1968.1098829

Lews, F., Syrmos, V., 1995. Optimal Control. Wiley, New Jersey, USA.

Luo, B., Wu, H.N., Huang, T., *et al.*, 2014. Data-based approximate policy iteration for affine nonlinear continuous-time optimal control design. *Automatica*, **50**(12):3281-3290. https://doi.org/10.1016/j.automatica.2014.10.056

Luo, B., Huang, T., Wu, H.N., *et al.*, 2015a. Data-driven H_∞ control for nonlinear distributed parameter systems. *IEEE Trans. Neur. Netw. Learn. Syst.*, **26**(11):2949-2961. https://doi.org/10.1109/TNNLS.2015.2461023

Luo, B., Wu, H.N., Huang, T., 2015b. Off-policy reinforcement learning for H_∞ control design. *IEEE Trans. Cybern.*, **45**(1):65-76. https://doi.org/10.1109/TCYB.2014.2319577

Luo, B., Wu, H.N., Li, H.X., 2015c. Adaptive optimal control of highly dissipative nonlinear spatially distributed processes with neuro-dynamic programming. *IEEE Trans. Neur. Netw. Learn. Syst.*, **26**(4):684-696. https://doi.org/10.1109/TNNLS.2014.2320744

Markman, J., Katz, I.N., 2000. An iterative algorithm for solving Hamilton-Jacobi type equations. *SIAM J. Sci. Comput.*, **22**(1):312-329. https://doi.org/10.1137/S1064827598344315

Sakamoto, N., van der Schaft, A.J., 2008. Analytical approximation methods for the stabilizing solution of the Hamilton-Jacobi equation. *IEEE Trans. Autom. Contr.*, **53**(10):2335-2350. https://doi.org/10.1109/TAC.2008.2006113

Saridis, G.N., Lee, C.S.G., 1979. An approximation theory of optimal control for trainable manipulators. *IEEE Trans. Syst. Man Cybern.*, **9**(3):152-159. https://doi.org/10.1109/TSMC.1979.4310171

Smears, I., Süli, E., 2014. Discontinuous Galerkin finite element approximation of Hamilton-Jacobi-Bellman equations with Cordes coefficients. *SIAM J. Numer. Anal.*, **52**(2):993-1016. https://doi.org/10.1137/130909536

Wu, H.N., Luo, B., 2012. Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear control. *IEEE Trans. Neur. Netw. Learn. Syst.*, **23**(12):1884-1895. https://doi.org/10.1109/TNNLS.2012.2217349

Yu, J., Jiang, Z.P., 2015. Global adaptive dynamic programming for continuous-time nonlinear systems. *IEEE Trans. Autom. Contr.*, **60**(11):2917-2929. https://doi.org/10.1109/TAC.2015.2414811