

## A FEEDBACK DESIGN FOR NUMERICAL SOLUTION TO OPTIMAL CONTROL PROBLEMS BASED ON HAMILTON-JACOBI-BELLMAN EQUATION

ZHEN-ZHEN TAO

School of Mathematics and Statistics, Beijing Institute of Technology  
Beijing 100081, China

BING SUN\*

School of Mathematics and Statistics, Beijing Institute of Technology  
Beijing 100081, China  
Beijing Key Laboratory on MCAACI, Beijing Institute of Technology  
Beijing 100081, China

**ABSTRACT.** In this paper, we present a feedback design for numerical solution to optimal control problems, which is based on solving the corresponding Hamilton-Jacobi-Bellman (HJB) equation. An upwind finite-difference scheme is adopted to solve the HJB equation under the framework of the dynamic programming viscosity solution (DPVS) approach. Different from the usual existing algorithms, the numerical control function is interpolated in turn to gain the approximation of optimal feedback control-trajectory pair. Five simulations are executed and both of them, without exception, output the accurate numerical results. The design can avoid solving the HJB equation repeatedly, thus efficaciously promote the computation efficiency and save memory.

**1. Introduction.** It is well-known that the optimal feedback control can be synthesized by the dynamic programming (DP) approach, while finding the optimal control in feedback form is generally argued as the Holy Grail of control theory ([16]). However, before we arrive at the feedback control by this approach, there are two difficulties that need to face. The notion of viscosity solution to partial differential equation (PDE) overcomes the first one that the HJB equation usually has no classical solution regardless of the smoothness of its coefficients ([7, 8, 9]). Under this framework, such kind of weak solution simultaneously holds the existence and uniqueness. The value function of the optimal control problem is the unique viscosity solution to the HJB equation. Nevertheless, for most of optimal control problems, it is usually impossible to find the analytical viscosity solution, which is the second difficulty to hinder the utilization of the DP approach. The numerical solution is almost the only practically significant way for finding the optimal control via solving the HJB equation. Luckily enough, nowadays there are some literatures

---

2020 *Mathematics Subject Classification.* 49L25, 49M30, 49N35, 65D05.

*Key words and phrases.* Dynamic programming, viscosity solution, interpolation, optimal feedback control, upwind finite difference scheme, Hamilton-Jacobi-Bellman equation.

The second author is supported in part by the National Natural Science Foundation of China under Grant No. 11471036.

\* Corresponding author: Bing Sun.

available devoted to these topics where the analysis of DP methods for deterministic and stochastic control problems is presented in detail ([3, 6, 11, 12]).

Besides the Bellman DP principle, another milestone work in control theory is contributed to the Pontryagin maximum principle, which establishes a necessary optimality condition satisfied by the investigational optimal control systems consisted of the state equation and its adjoint system ([23, 26]). Nevertheless, in essence, the control presented by the Pontryagin maximum principle is of open-loop, which is usually less than satisfactory. Moreover, it is notorious for its sophistication and the fact that it rarely gives an insight into the structure of the optimal controls ([24]). Generally speaking, a feedback control system is considered superior to an open-loop system. Many unnecessary disturbances and noise signals from outside the system can be rejected by using feedback ([4]). Furthermore, even from the numerical solution point of view, for the two-point boundary problem obtained by the necessary condition of optimality, the recognized shooting method has to overcome the difficulty of “guess” for the initial data to start the iterative numerical process ([14, 27]), which is generally not an easy job.

People need to make great efforts to develop more efficient algorithms to find the optimal control. Two real reasons for this could not be clearer. One is due to the lack of the analytical solution to the practical optimal control problems. We must numerically solve the investigational problem. Furthermore, the ultimate aim of the control theory is to use it in the real world. It also coincides with the requirement of the information age. The other reason is caused by the curse-of-dimensionality, the inherent defect of the DP approach ([22]). Thanks to many model reduction techniques ([1, 3, 13, 20, 19]), this problem can be partially circumvented. It makes some effective algorithms for low dimensional problems, thus initially confined to only “toy” problems, can be generalized to deal with much more complicated practical optimal control problems. The upwind finite-difference scheme is such a well-adapted algorithm and has been successfully applied to many examples ([13, 14, 15]). And most of all, its convergence has been rigorously proven in [28].

Certainly it is not easy to find an optimal feedback control by the HJB-based approach and lots of complicated computations are usually involved. Even so, on this topic, there are still many interesting results available in literature, such as [1, 5, 17, 19, 22, 29], to name just a few. We can unfortunately select only some more relevant research to mention here. For two practical optimal control problems, [14, 15] obtain the optimal feedback controls by repeatedly calling the algorithm to the HJB equation. In [2, 10, 18], the HJB equation is firstly solved to gain the value function. And then the convex combination of numerical value function is applied to obtain the approximation of optimal feedback control. Note that the obtained numerical solution of control function is completely ignored and not directly applied to obtain optimal feedback control. A class of explicit Markov chain approximation methods are introduced and studied in detail in [21]. And these methods are tailored to solving the HJBs for both stochastic and deterministic continuous time nonlinear optimal control problems on finite and infinite horizons. The paper [20] considers a distributed volume control problem in infinite time region and obtains the optimal control by the convex combination of the control values at each point of the polyhedron, which can be interpreted as a special interpolation. The idea of using the interpolation to find the optimal control is also mentioned in

[13] but it is kept mostly to one-dimensional linear interpolation for one-dimensional problem. The multi-dimensional problems are not involved in the analysis.

In this paper, we are concerned with a feedback design for numerical solution to the optimal control problem in finite time horizon under the framework of the DPVS approach. To synthesize the optimal feedback control, we solve the HJB equation by the upwind finite-difference scheme. Different from the usual existing algorithms, the numerical control function is interpolated in turn to gain the approximation of optimal feedback control-trajectory pair, which avoid solving the HJB equation repeatedly, thus efficaciously promote the computation efficiency. Moreover, the design can deal with the optimal control problem equipped with multiple controls. And more generally, here, the adopted interpolation is not limited to one-dimensional linear interpolation. Other interpolation techniques including multi-dimensional linear interpolation, cubic spline interpolation, nearest-neighbor interpolation and cubic-Hermite interpolation go well likewise in this new design.

The remainder of this paper is organized as follows. In Section 2, we revisit the procedure of the DPVS approach and present the upwind finite-difference scheme to the HJB equation. An interpolation algorithm then follows, which consists of the core of this paper. The effectiveness of the algorithm is proven in Section 3 by solving five quite distinct examples from each other before some concluding comments are made in Section 4.

**2. DPVS approach and interpolation algorithm.** Set  $T > 0$  and  $U \subset \mathbb{R}^m$  be a closed, bounded and convex set. Denote by  $\mathcal{U}[0, T] = L^\infty([0, T]; U)$  the admissible control set. We consider the following control system

$$\begin{cases} \frac{d}{dt}y(t) = f(t, y(t), u(t)), & t \in (0, T], \\ y(0) = y_0, \end{cases} \quad (2.1)$$

where  $y : [0, T] \rightarrow \mathbb{R}^n$  is the state, the control  $u(\cdot) \in \mathcal{U}[0, T]$  and  $y_0 \in \mathbb{R}^n$  is a given initial value. Assume that the mapping  $f : [0, T] \times \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$  is continuous. Take the cost functional as

$$J(u(\cdot)) = \int_0^T L(s, y(s), u(s))ds + h(y(T)), \quad (2.2)$$

where both the running cost  $L : [0, T] \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  and the terminal cost  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  are continuous. The optimal control problem is to minimize (2.2) over  $\mathcal{U}[0, T]$ , i.e.,

$$\inf_{u \in \mathcal{U}[0, T]} J(u(\cdot)) \quad (2.3)$$

subject to the control system (2.1).

In order to present the DP principle, we consider a family of dynamic optimal control problems. Namely, over  $\mathcal{U}[t, T] = L^\infty([t, T]; U)$ , we minimize the cost functional

$$J(u(\cdot); t, x) = \int_t^T L(s, y(s), u(s))ds + h(y(T)),$$

subject to

$$\begin{cases} \frac{d}{ds}y(s) = f(s, y(s), u(s)), & s \in (t, T], \\ y(t) = x, \end{cases}$$

where  $(t, x) \in [0, T] \times \mathbb{R}^n$  is the initial time and state. Define the value function by

$$\begin{aligned} v(t, x) &= \inf_{u(\cdot) \in \mathcal{U}[t, T]} J(u(\cdot); t, x), \quad (t, x) \in [0, T] \times \mathbb{R}^n, \\ v(T, x) &= h(x), \quad x \in \mathbb{R}^n, \end{aligned} \quad (2.4)$$

which is the associated minimal cost functional. It satisfies the DP principle ([3])

$$v(t, x) = \inf_{u(\cdot) \in \mathcal{U}[t, \tau]} \left\{ \int_t^\tau L(s, y(s; t, x), u(s)) ds + v(\tau, y(\tau; t, x)) \right\},$$

for any  $\tau \in [t, T]$ .

Then, if the value function  $v$  is smooth enough, such as  $v \in C^1([0, T] \times \mathbb{R}^n)$ , the following HJB equation can be derived from the DP principle

$$\begin{aligned} -\frac{\partial}{\partial t} v(t, x) - \inf_{u \in U} \{ f(t, x, u) \cdot \nabla_x v(t, x) + L(t, x, u) \} &= 0, \quad (t, x) \in [0, T] \times \mathbb{R}^n, \\ v(T, x) &= h(x), \quad x \in \mathbb{R}^n, \end{aligned} \quad (2.5)$$

where  $\nabla_x$  denotes the gradient with respect to  $x$ . By solving the HJB equation (2.5), we next turn to the feedback synthesis, i.e., the construction of the optimal control  $u^*(t)$  with the optimal state  $y^*(t)$ . For any mapping  $\bar{u}(t, x)$  satisfying

$$\bar{u}(t, x) \in \arg \inf_{u \in U} \{ f(t, x, u) \cdot \nabla_x v(t, x) + L(t, x, u) \},$$

we set

$$u^*(t) = \bar{u}(t, y^*(t)), \quad (2.6)$$

for almost all  $t \in [0, T]$ , where  $y^*(t)$  satisfies

$$\frac{d}{dt} y^*(t) = f(t, y^*(t), \bar{u}(t, y^*(t))), \quad t \in (0, T], \quad (2.7)$$

with  $y^*(0) = y_0$ . Then we have

$$\begin{aligned} & f(t, y^*(t), u^*(t)) \cdot \nabla_x v(t, y^*(t)) + L(t, y^*(t), u^*(t)) \\ &= \inf_{u \in U} \{ f(t, y^*(t), u) \cdot \nabla_x v(t, y^*(t)) + L(t, y^*(t), u) \} \end{aligned}$$

for almost all  $t \in [0, T]$ . Thus  $u^*(\cdot)$  is an optimal control of (2.3) from [3]. Here,  $\bar{u}$  in (2.6) is called the control function or the optimal feedback mapping. When the mapping  $\bar{u}$  is known in (2.7), the system (2.7) is a closed loop system.

The DPVS approach tells us that once a viscosity solution of the HJB equation (2.5) is obtained numerically, we are able to find the numerical solution of the feedback law by the feedback synthesis above.

Next we revisit the upwind finite-difference scheme to the HJB equation (2.5). Actually, it is such a well-adapted algorithm and has been successfully applied to many examples ([14, 13, 15]). And most of all, its convergence has been rigorously proven in [28].

Take a polygonal region  $[0, T] \times \prod_{k=1}^n [a_k, b_k]$  as the computational domain. Set the time step  $\Delta t = \frac{T}{N}$  and denote  $t_j = T + j\Delta t$ ,  $j = 0, 1, 2, \dots, N$ , which divide the time domain  $[0, T]$  into  $N$  equal parts. Similarly, for each  $k = 1, 2, \dots, n$ , divide  $[a_k, b_k]$  into  $M_k$  equal parts. Denote  $\Delta x_k = \frac{b_k - a_k}{M_k}$ ,  $k = 1, 2, \dots, n$ , the space steps. And let  $i = (i_1, i_2, \dots, i_n)$ ,  $ik^+ = (i_1, \dots, i_{k-1}, i_k + 1, i_{k+1}, \dots, i_n)$ ,  $ik^- = (i_1, \dots, i_{k-1}, i_k - 1, i_{k+1}, \dots, i_n)$ , where  $i_k = 0, 1, \dots, M_k$ ,  $k = 1, 2, \dots, n$ . Denote

the point  $x_i = (a_1 + i_1 \Delta x_1, a_2 + i_2 \cdot \Delta x_2, \dots, a_n + i_n \Delta x_n)$ . Then the upwind finite-difference scheme of the HJB equation (2.5) can be written as

$$\begin{aligned} \frac{v_i^{j+1} - v_i^j}{\Delta t} + \sum_{k=1}^n \left( \frac{1 + \text{sign} f_{k,i}^j}{2} \frac{v_{i_{k+}}^j - v_i^j}{\Delta x_k} + \frac{1 - \text{sign} f_{k,i}^j}{2} \frac{v_i^j - v_{i_{k-}}^j}{\Delta x_k} \right) f_{k,i}^j + L_i^j &= 0, \\ u_i^{j+1} = \arg \inf_{u \in U} \left( \sum_{k=1}^n f_k(t_{j+1}, x_i, u) \frac{v_i^{j+1} - v_{i_{k-}}^{j+1}}{\Delta x_k} + L(t_{j+1}, x_i, u) \right), \end{aligned} \quad (2.8)$$

for  $i_k = j + 1, j + 2, \dots, M_k - j - 1$ ,  $k = 1, 2, \dots, n$  and  $j = 0, 1, \dots, N - 1$ . Here, the sign  $z$  denotes the sign of  $z$ . And  $u_i^j, v_i^j$  are, respectively, the approximations of  $\bar{u}(t_j, x_i)$  and  $v(t_j, x_i)$ . Then, denote

$$f_{k,i}^j = f_k(t_j, x_i, u_i^j), \quad L_i^j = L(t_j, x_i, u_i^j).$$

In the following, we construct a HJB-based feedback algorithm for (2.3). Its detailed steps are listed as follows.

**Step 1. Initial partition on time and space.** Divide the calculation domain into mesh as mentioned in Section 2 above. Denote  $(u^j, y^j)$  by the optimal control-trajectory pair at  $t^j = -j\Delta t$  for  $j = 0, 1, \dots, N$ .

**Step 2. Compute the approximations of value function and control function.** Based on the numerical scheme (2.8), set  $\gamma_k = \frac{\Delta t}{\Delta x_k}$  for  $k = 1, 2, \dots, n$ .

**Sub-step 2.1.** By the same space steps  $\Delta x_k$ ,  $k = 1, 2, \dots, n$ , extend the initial solution region  $\prod_{k=1}^n [a_k, b_k]$  to a new one  $\prod_{k=1}^n [a_k - (N + 1)\Delta x_k, b_k + (N + 1)\Delta x_k]$  by adding the  $2(N + 1)$  nodes. (This is a rough expansion and more precise extension steps can be found in [29]). Renumber the points on the new region  $\prod_{k=1}^n [a_k - (N + 1)\Delta x_k, b_k + (N + 1)\Delta x_k]$  like in Section 2. Denote  $ii = (ii_1, ii_2, \dots, ii_n)$ ,  $ii_{k+} = (ii_1, \dots, ii_{k-1}, ii_k + 1, ii_{k+1}, \dots, ii_n)$ ,  $ii_{k-} = (ii_1, \dots, ii_{k-1}, ii_k - 1, ii_{k+1}, \dots, ii_n)$ , where  $ii_k = 0, 1, \dots, M_k + 2(N + 1)$  and  $k = 1, 2, \dots, n$ . Define  $x_{ii} = (a_1 + (ii_1 - N - 1)\Delta x_1, a_2 + (ii_2 - N - 1)\Delta x_2, \dots, a_n + (ii_n - N - 1)\Delta x_n)$ .

**Sub-step 2.2.** Initialize  $\{v_{ii}^0\}_{ii_k=0}^{M_k+2(N+1)}$  and  $\{u_{ii}^0\}_{ii_k=0}^{M_k+2(N+1)}$  as follows.

$$\begin{aligned} v_{ii}^0 &= h(x_{ii}), \\ u_{ii}^0 &= \arg \inf_{u \in U} \left( \sum_{k=1}^n f_k(t_0, x_{ii}, u) \frac{v_{ii}^0 - v_{ii_{k-}}^0}{\Delta x_k} + L(t_0, x_{ii}, u) \right). \end{aligned}$$

**Sub-step 2.3.** For  $j = 0$  to  $N - 1$ , compute  $v_{ii}^{j+1}$  and  $u_{ii}^{j+1}$  as follows. For  $ii_k = j + 1$  to  $M_k + 2N - j + 1$ ,  $k = 1, 2, \dots, n$ , set

$$\begin{aligned} v_{ii}^{j+1} &= \left( 1 + \sum_{k=1}^n \gamma_k |f_{k,ii}^j| \right) v_{ii}^j - \sum_{k=1}^n \frac{1 + \text{sign} f_{k,ii}^j}{2} \gamma_k f_{k,ii}^j v_{ii_{k+}}^j \\ &\quad + \sum_{k=1}^n \frac{1 - \text{sign} f_{k,ii}^j}{2} \gamma_k f_{k,ii}^j v_{ii_{k-}}^j - \Delta t L_{ii}^j, \\ u_{ii}^{j+1} &= \arg \inf_{u \in U} \left( \sum_{k=1}^n f_k(t_{j+1}, x_{ii}, u) \frac{v_{ii}^{j+1} - v_{ii_{k-}}^{j+1}}{\Delta x_k} + L(t_{j+1}, x_{ii}, u) \right). \end{aligned}$$

**Sub-step 2.4.** Choose the numerical solution of value function and control function over  $[0, T] \times \prod_{k=1}^n [a_k, b_k]$ . Denote them by  $\left\{ \{v_i^j\}_{i_k=0}^{M_k} \right\}_{j=0}^N$  and  $\left\{ \{u_i^j\}_{i_k=0}^{M_k} \right\}_{j=0}^N$ ,  $k = 1, 2, \dots, n$ , respectively. Here,  $v_i^j$  is the approximation of value function  $v(t_j, x_i)$  and  $u_i^j$  is the approximation of control function  $\bar{u}(t_j, x_i)$ .

**Step 3. Compute the optimal pairs  $\{(u^j, y^j)\}_{j=0}^N$ .** For initial setting, let  $y^0 = y_0$ . Set  $j = 0$ .

**Sub-step 3.1.** Calculate the  $j$ -th optimal feedback control  $u^j$  by applying the numerical solution  $\left\{ u_i^{N-j} \right\}_{i_k=0}^{M_k}$ ,  $k = 1, 2, \dots, n$ , obtained in Step 2. Choose the linear interpolation (or cubic spline interpolation, nearest-neighbor interpolation and cubic-Hermite interpolation) for point  $y^j$  by using the value  $\left\{ u_i^{N-j} \right\}_{i_k=0}^{M_k}$  at known points  $\{x_i\}_{i_k=0}^{M_k}$ ,  $k = 1, 2, \dots, n$ . Actually, the value obtained by interpolation is an approximation of  $\bar{u}(t^j, y^j)$ , which is denoted as  $\tilde{u}(t^j, y^j)$ . Then, we set

$$u^j = \tilde{u}(t^j, y^j).$$

**Sub-step 3.2.** Calculate the  $(j+1)$ -th optimal trajectory  $y^{j+1}$ . Discrete the state equation by the forward difference:

$$y^{j+1} = y^j - f(t^j, y^j, u^j)\Delta t.$$

With the known  $u^j$  and  $y^j$ , we solve the finite difference equation above to obtain the  $(j+1)$ -th optimal trajectory  $y^{j+1}$ .

**Sub-step 3.3.** Iterate for the next time instant. Set  $j = j + 1$ . If  $j > N$ , end the procedure. Otherwise, go to Sub-step 3.1. Continue the iteration until we get all optimal feedback control-trajectory pairs  $\{(u^j, y^j)\}_{j=0}^N$ .

**Remark 1.** In Sub-step 3.1, we can also adopt other interpolation methods, such as polynomial interpolation, average interpolation, the Newton interpolation, Lagrange's interpolation and so on ([25]). Similarly, other difference method can be used to discrete state equation in Sub-step 3.2.

**3. Simulations.** This section is devoted to five numerical examples to verify the effectiveness of the algorithm. The simulations are implemented by using MATLAB programme in a desktop computer with a core process (i5-6600M) and 8GB of random access memory.

**Example 1.** Consider the following control problem ([31])

$$\begin{cases} \dot{y}(t) = y(t)u(t), & t \in (0, 1], \\ y(0) = y_0, \\ J(u(\cdot)) = -y(1), \\ \min_{u(\cdot) \in \mathcal{U}} J(u(\cdot)), \end{cases}$$

where  $\mathcal{U} = L^\infty([0, 1]; U)$ ,  $U = [0, 1]$ ,  $y_0 \in \mathbb{R}$  is a given initial value. Its corresponding HJB equation is

$$-\frac{\partial}{\partial t}v(t, x) - \inf_{u \in [0,1]} \{xu \cdot \nabla_x v(t, x)\} = 0, \quad (t, x) \in [0, 1] \times \mathbb{R},$$

$$v(1, x) = -x, \quad x \in \mathbb{R}.$$

Its exact value function  $v$  is

$$v(t, x) = \begin{cases} -xe^{1-t}, & \text{if } x > 0, \\ -x, & \text{if } x \leq 0, \end{cases}$$

and the control function is

$$\bar{u}(t, x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x \leq 0. \end{cases} \quad (3.1)$$

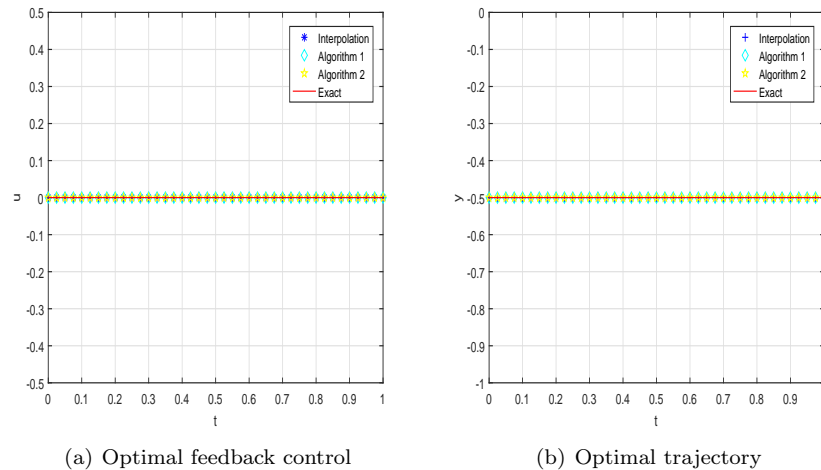


FIGURE 1. Numerical and exact solutions in Example 1 with initial state  $y_0 = -\frac{1}{2}$ .

Here, we choose the region  $D = [0, 1] \times [-\frac{3}{2}, \frac{3}{2}]$ , i.e.,  $0 \leq t \leq 1$  and  $-\frac{3}{2} \leq x \leq \frac{3}{2}$ . The number of nodes  $N$  on the  $t$  axis is chosen as 41. And the number of nodes  $M$  on the  $x$  axis is set as 31. Then the time step  $\Delta t = \frac{1}{40}$  and the space step  $\Delta x = \frac{1}{10}$ . Adopting the linear interpolation, we numerically solve this example by the interpolation algorithm presented in Section 2 above. And the results are plotted in Figures 1 and 2.

Moreover, in order to show the performance of the presented interpolation algorithm, two existing algorithms are additionally chosen to solve Example 1, simultaneously. The corresponding results are also given in Figures 1 and 2. And the comparisons on numerical simulations are listed in Table 1.

Actually, one chosen algorithm is the minimisation of the Hamiltonian. People can refer to Algorithm 3.2 in [13] for details. For convenience, we use Algorithm 1 to denote it. In the other one algorithm, the optimal feedback control is also gained by minimizing the Hamiltonian function. But it is different from Algorithm 1, in which the value function is obtained by the convex combination of numerical value function ([18]). We name it as Algorithm 2.

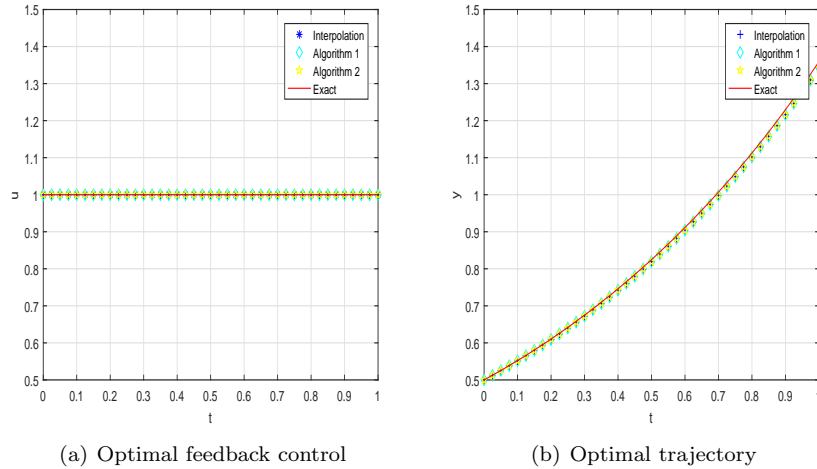


FIGURE 2. Numerical and exact solutions in Example 1 with initial state  $y_0 = \frac{1}{2}$ .

When  $y_0 = -\frac{1}{2}$ , it is easy to obtain the optimal control  $u^*(t) \equiv 0, t \in [0, 1]$ , and the optimal trajectory  $y^*(t) \equiv -\frac{1}{2}, t \in [0, 1]$ . Figure 1(a) shows the numerical solutions of optimal feedback control agrees well with the exact solution for these three algorithms. In Figure 1(b), the numerical solution of optimal trajectory also agrees well with its exact solution. When  $y_0 = \frac{1}{2}$ , the optimal control is  $u^*(t) \equiv 1, t \in [0, 1]$ , and the optimal trajectory is  $y^*(t) = \frac{1}{2}e^t, t \in [0, 1]$ . Figure 2 shows both the numerical solutions of optimal feedback control and trajectory agree well with the exact solutions when initial state  $y_0 = \frac{1}{2}$ .

For these three algorithms, Table 1 lists the computed errors in the maximum norm between the numerical solutions and exact solutions including the optimal feedback control  $u^*$  and optimal trajectory  $y^*$  with initial state  $y_0 = -\frac{1}{2}$  and  $y_0 = \frac{1}{2}$ . So small error estimates tell us that the numerical solutions obtained by the interpolation algorithm, Algorithms 1 and 2 are very close to the exact solutions.

Moreover, the CPU times are listed in Table 1. They are the computing time of these three algorithms after we obtain the numerical control function and numerical value function by upwind finite-difference method. By comparing the CPU times, we find that the interpolation algorithm takes the least time than Algorithms 1 and 2 with the similar error estimation. It illustrates that linear interpolation is meaningful for finding an approximation of the optimal feedback control, although the control function (3.1) is discontinuous. In all, numerical results shows that the interpolation algorithm is not only feasible but also efficient for gaining the optimal feedback control.

**Example 2.** Consider the following control problem ([30])

$$\begin{cases} \dot{y}(t) = y(t) + u(t), & t \in (0, 1], \\ y(0) = 1, \\ J(u(\cdot)) = \frac{1}{2} \int_0^1 u^2(t) dt + \frac{1}{2}y^2(1), \\ \min_{u(\cdot) \in \mathcal{U}} J(u(\cdot)), \end{cases}$$



Algorithm	Initial state	Control	Trajectory	CPU(s)
Interpolation	$y_0 = -\frac{1}{2}$	6.6000e-05	3.3001e-05	0.005615
	$y_0 = \frac{1}{2}$	6.6000e-05	0.016695	0.005484
Algorithm 1	$y_0 = -\frac{1}{2}$	6.6000e-05	3.3001e-05	0.015370
	$y_0 = \frac{1}{2}$	6.6000e-05	0.016695	0.016507
Algorithm 2	$y_0 = -\frac{1}{2}$	6.6000e-05	3.3001e-05	0.025182
	$y_0 = \frac{1}{2}$	6.6000e-05	0.016695	0.025733

TABLE 1. The maximum norm error estimates and CPU times for Example 1 with initial state  $y_0 = -\frac{1}{2}$  and  $y_0 = \frac{1}{2}$ , respectively.

where  $\mathcal{U} = L^\infty([0, 1]; U)$ ,  $U = \mathbb{R}$ . Its corresponding HJB equation can be read as

$$-\frac{\partial}{\partial t}v(t, x) - \inf_{u \in \mathbb{R}} \left\{ (x + u) \cdot \nabla_x v(t, x) + \frac{1}{2}u^2 \right\} = 0, \quad (t, x) \in [0, 1] \times \mathbb{R}, \quad (3.2)$$

$$v(1, x) = \frac{1}{2}x^2, \quad x \in \mathbb{R}.$$

The exact solution to the HJB equation (3.2) is

$$v(t, x) = \frac{x^2}{1 + e^{2(t-1)}},$$

and the control function is

$$\bar{u}(t, x) = \frac{-2x}{1 + e^{2(t-1)}}. \quad (3.3)$$

With (2.6), (2.7) and (3.3), we obtain the optimal feedback control  $u^*(t) = \frac{-2e^{-t}}{1+e^{-2}}$  and the optimal trajectory  $y^*(t) = \frac{e^{-t} + e^{t-2}}{1+e^{-2}}$ .

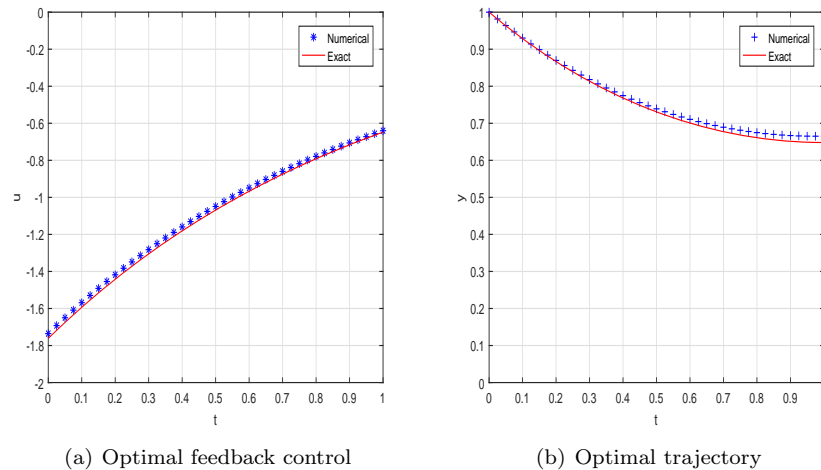


FIGURE 3. Numerical and exact solutions in Example 2 with initial state  $y_0 = 1$ .

Now we numerically solve Example 2. Choose the computational region  $[0, 1] \times [0, 1]$ , i.e.,  $0 \leq t \leq 1$  and  $0 \leq x \leq 1$ . Here we choose a uniform mesh along  $t$  axis with  $N = 41$  nodes, i.e.,  $\Delta t = \frac{1}{40}$ , and a uniform mesh along  $x$  axis with  $M = 21$  nodes, i.e.,  $\Delta x = \frac{1}{20}$ . In this example, the chosen interpolation is also the linear interpolation. The obtained numerical solutions are plotted in Figure 3. By this figure, we see that both the numerical solutions of optimal feedback control and trajectory agree well with the exact solutions. Moreover, in Table 2, we list the maximum errors between the numerical solutions and exact solutions of optimal feedback control and trajectory. They are, respectively, 0.025896 and 0.017297, which are very small. It also shows that the algorithm is effective.

Error	Control	Trajectory
$y_0 = 1$	0.025896	0.017297

TABLE 2. The maximum norm error estimates for Example 2 with initial state  $y_0 = 1$ .

In addition, in Examples 1 and 2, the control problems are all one-dimensional problems. Next, we present other two optimal control examples with two space dimensions.

**Example 3.** Consider the following two-dimensional control problem ([29]) with two state variables and one control.

$$\begin{cases} \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} u(t), & t \in (0, 1], \\ x(0) = x_0, & y(0) = y_0, \\ J(u(\cdot)) = -x(1) - y(1), \\ \min_{u(\cdot) \in \mathcal{U}} J(u(\cdot)), \end{cases}$$

where  $\mathcal{U} = L^\infty([0, 1]; U)$ ,  $U = [0, 1]$ ,  $(x_0, y_0) \in \mathbb{R}^2$ . Its corresponding HJB equation is

$$\begin{aligned} -\frac{\partial}{\partial t} v(t, x, y) - \inf_{u \in [0, 1]} \{xu \cdot v_x(t, x, y) + yu \cdot v_y(t, x, y)\} &= 0, \quad (t, x, y) \in [0, 1] \times \mathbb{R}^2, \\ v(1, x, y) &= -(x + y), \quad (x, y) \in \mathbb{R}^2. \end{aligned} \tag{3.4}$$

In this example, the exact solution of the value function is

$$v(t, x, y) = \begin{cases} -(x + y)e^{1-t}, & \text{if } x + y > 0, \\ -(x + y), & \text{if } x + y \leq 0, \end{cases}$$

and the control function is

$$\bar{u}(t, x, y) = \begin{cases} 1, & \text{if } x + y > 0, \\ 0, & \text{if } x + y \leq 0. \end{cases} \tag{3.5}$$

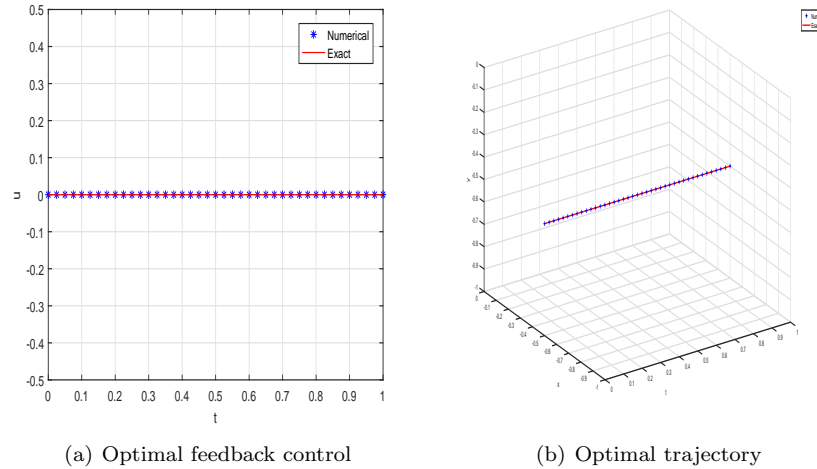


FIGURE 4. Numerical and exact solutions in Example 3 with initial states  $(x_0, y_0) = (-\frac{1}{2}, -\frac{1}{2})$ .

From (2.6), (2.7) and (3.5), when initial states  $(x_0, y_0) = (-\frac{1}{2}, -\frac{1}{2})$ , the optimal feedback control is  $u^*(t) \equiv 0$ ,  $t \in [0, 1]$  and the optimal trajectory is  $(x^*(t), y^*(t)) \equiv (-\frac{1}{2}, -\frac{1}{2})$ ,  $t \in [0, 1]$ . When the initial states  $(x_0, y_0) = (\frac{1}{2}, \frac{1}{2})$ , the optimal feedback control is  $u^*(t) \equiv 1$  and the optimal trajectory is  $(x^*(t), y^*(t)) = (\frac{1}{2}e^t, \frac{1}{2}e^t)$ ,  $t \in [0, 1]$ .

Next we use the interpolation algorithm to solve the HJB equation (3.4) in the range  $[0, 1] \times [-\frac{3}{2}, \frac{3}{2}]^2$ , i.e.,  $0 \leq t \leq 1$ ,  $-\frac{3}{2} \leq x \leq \frac{3}{2}$ ,  $-\frac{3}{2} \leq y \leq \frac{3}{2}$ . Here we still choose a uniform mesh on  $t$  axis,  $x$  axis and  $y$  axis. The nodes  $N$  on  $t$  axis is set as 41, i.e.,  $\Delta t = \frac{1}{40}$ . The nodes on  $x$  and  $y$  axis are set as  $M_1 = M_2 = 31$ , i.e.,  $\Delta x = \Delta y = \frac{1}{10}$ . Again, the linear interpolation is chosen to be applied in the algorithm.

From Figures 4 and 5, we see that, even with different initial values, the images of numerical solutions agree well with the exact solutions for the optimal feedback control and trajectory.

Example 3 shows that, even for two-dimensional control problem, the linear interpolation is meaningful for finding an approximation of the optimal feedback control. On the other hand, we find that the discontinuity of the control function (3.5) does not hinder us to find the optimal feedback control.

Errors	Control	Trajectory
$(x_0, y_0) = (-\frac{1}{2}, -\frac{1}{2})$	6.6000e-05	4.6671e-05
$(x_0, y_0) = (\frac{1}{2}, \frac{1}{2})$	6.6000e-05	0.023611

TABLE 3. The maximum norm error estimates for Example 3 with initial states  $(x_0, y_0) = (-\frac{1}{2}, -\frac{1}{2})$  and  $(x_0, y_0) = (\frac{1}{2}, \frac{1}{2})$ , respectively.

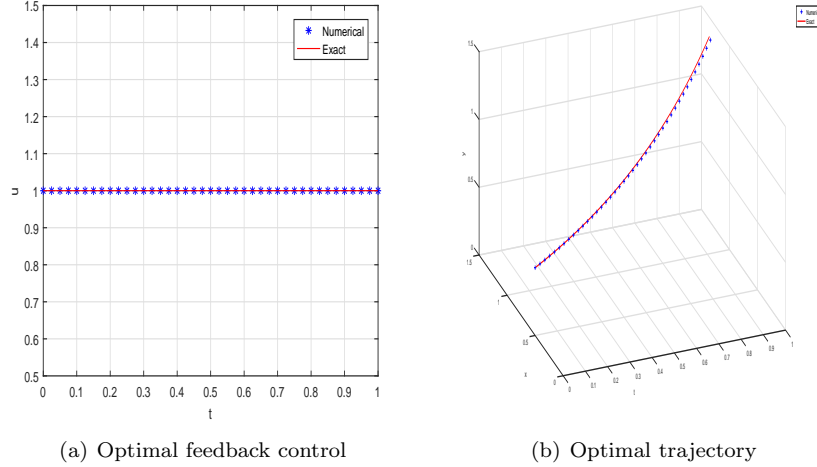


FIGURE 5. Numerical and exact solutions in Example 3 with initial states  $(x_0, y_0) = (\frac{1}{2}, \frac{1}{2})$ .

Furthermore, all three examples above are equipped with one control. And only the linear interpolation is adopted in the algorithm. And to show the effectiveness and efficiency of the interpolation algorithm in an even better fashion, we consider an optimal control problem with two states and two controls in Example 4. And four interpolations including linear interpolation, cubic spline interpolation, nearest-neighbor interpolation and cubic-Hermite interpolation are successively adopted in the investigation.

**Example 4.** Consider the following optimal control problem with two state variables and two controls.

$$\begin{cases} \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = \begin{pmatrix} x_1(t) + u_1(t) \\ u_2(t) \end{pmatrix}, & t \in (0, 1], \\ x_1(0) = x_{10}, & x_2(0) = x_{20}, \\ J(u_1(\cdot), u_2(\cdot)) = \frac{1}{2} \int_0^1 [u_1^2(t) + u_2^2(t) + x_2^2(t)] dx + \frac{1}{2} x_1^2(1), \\ \min_{(u_1(\cdot), u_2(\cdot)) \in \mathcal{U}} J(u_1(\cdot), u_2(\cdot)), \end{cases} \quad (3.6)$$

where  $(x_{10}, x_{20}) \in \mathbb{R}^2$ ,  $\mathcal{U} = L^\infty([0, 1]; U)$ ,  $U = \{(p, q) \in \mathbb{R}^2 \mid |p| \leq 1, |q| \leq 1, |p + q^2| \leq 1\}$ . Denote by  $|\cdot|$  the absolute value. The HJB equation of (3.6) can be written as

$$\begin{aligned} & -\frac{\partial}{\partial t} v(t, x_1, x_2) - \inf_{|u_1|, |u_2|, |u_1 + u_2^2| \leq 1} \left\{ (x_1 + u_1) \cdot v_{x_1}(t, x_1, x_2) \right. \\ & \left. + u_2 \cdot v_{x_2}(t, x_1, x_2) + \frac{1}{2} u_1^2 + \frac{1}{2} u_2^2 + \frac{1}{2} x_2^2 \right\} = 0, \quad (t, x_1, x_2) \in [0, 1) \times \mathbb{R}^2, \\ & v(1, x_1, x_2) = \frac{1}{2} x_1^2, \quad (x_1, x_2) \in \mathbb{R}^2. \end{aligned} \quad (3.7)$$

Here we choose the computational region  $[0, 1] \times [-1, 1]^2$ , i.e.  $0 \leq t \leq 1$ ,  $-1 \leq x_1 \leq 1$  and  $-1 \leq x_2 \leq 1$ . The number of nodes  $N$  on  $t$  axis is chosen as 41. And the number of nodes on  $x_1$  and  $x_2$  axis are set as  $M_1 = M_2 = 41$ . Then the time step  $\Delta t = \frac{1}{40}$  and the space steps  $\Delta x_1 = \Delta x_2 = \frac{1}{20}$ . To visualize the numerical result, we plot the value function  $v$ , two control functions  $\bar{u}_1$  and  $\bar{u}_2$ , as well as  $\bar{u}_1 + \bar{u}_2^2$  at different cross-sections in Figures 6-8. From Figures 6(b-d), 7(b-d) and 8(b-d), we see that both the numerical controls  $\bar{u}_1$  and  $\bar{u}_2$ , and  $\bar{u}_1 + \bar{u}_2^2$  satisfy the given constraints.

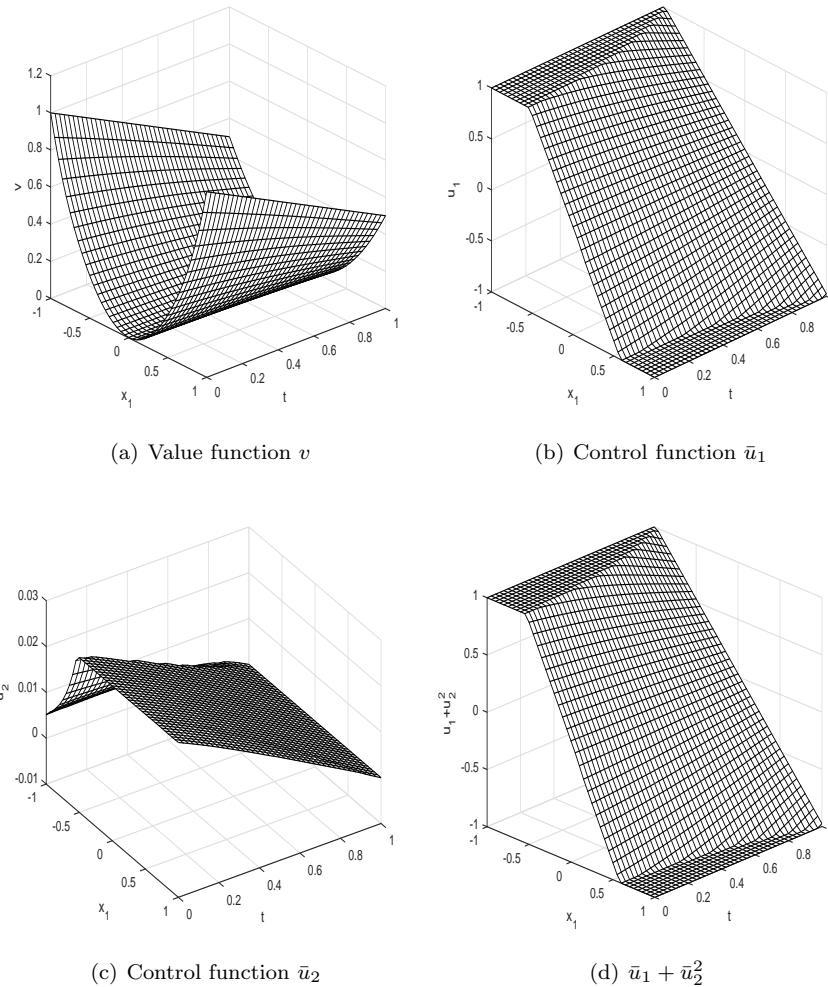


FIGURE 6. Numerical solutions of HJB equation (3.6) in Example 4.

Moreover, in this example, the exact value function  $v$  and control function  $(\bar{u}_1, \bar{u}_2)$  of the HJB equation (3.7) are unknown in literature. Although the control variables have so many constraints including nonlinear constraint, when  $(x_{10}, x_{20}) = (\frac{1}{2}, -\frac{1}{2})$ , by calculation we find the optimal controls  $u_1^*(t) = \frac{-e^{-t}}{1+e^{-2}}$ ,  $u_2^*(t) = \frac{e^{-t}-e^{t-2}}{2(1+e^{-2})}$ ,  $t \in [0, 1]$ , and the optimal trajectories  $x_1^*(t) = \frac{e^{-t}+e^{t-2}}{2(1+e^{-2})}$ ,  $x_2^*(t) = -\frac{e^{-t}+e^{t-2}}{2(1+e^{-2})}$ ,  $t \in [0, 1]$ .

Furthermore, in this example, we utilize the linear interpolation and the cubic spline interpolation to find the optimal feedback control. And their respective numerical results are plotted in Figures 9 and 10. From these two figures, we see that, no matter if it is the linear interpolation or the cubic spline interpolation, all numerical solutions have a good agreement with the exact solutions.

Then, in order to investigate the influence of different interpolation on the interpolation algorithm, we provide the error results of four interpolation cases for this example. In Table 4, we list the computational errors in the maximum norms between the numerical solutions and exact solutions including optimal control  $(u_1^*, u_2^*)$  and optimal trajectory  $(x_1^*, x_2^*)$  with initial state  $x_{10} = \frac{1}{2}$  and  $x_{20} = -\frac{1}{2}$ .

In the linear interpolation case, they are 0.038706 and 0.02681, respectively. As for the cubic spline interpolation case, they are 0.038706 and 0.026801, respectively. In nearest-neighbor and cubic-Hermite interpolation cases, we also get very small errors as well. Although the errors in nearest-neighbor case are little bit bigger than others, all of them show a nice approximation to the exact solutions. Relatively speaking, by Table 4, we find that the nearest-neighbor interpolation is the worst among these four interpolations. And other three interpolations almost perform equally well. From these small error estimates, we know that the numerical solutions obtained by the interpolation algorithm are very close the exact solutions. The experimental results of Example 4 demonstrate again that interpolation algorithm is extremely effective.

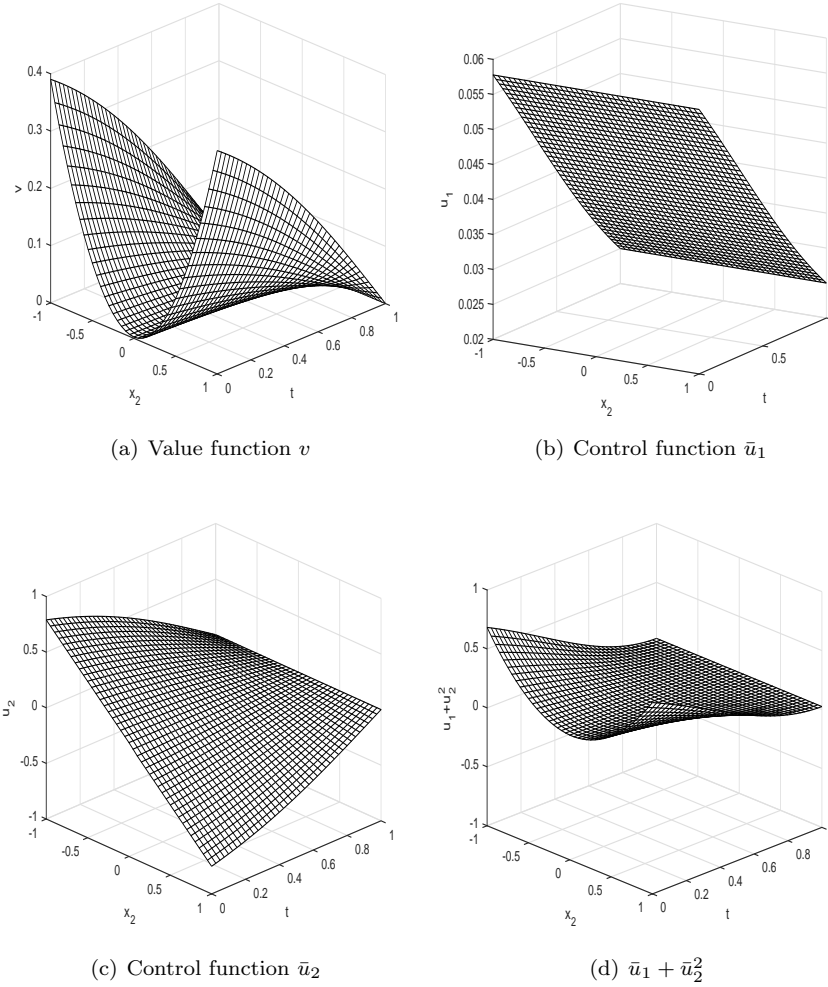
Error	Control	Trajectory
Linear	0.038706	0.02681
Cubic spline	0.038706	0.026801
Nearest-neighbor	0.069388	0.030588
Cubic-Hermite	0.038706	0.026804

TABLE 4. The maximum norm error estimates for Example 4 with initial states  $(x_{10}, x_{20}) = (\frac{1}{2}, -\frac{1}{2})$  in four interpolation cases.

Let me emphasize that, for four examples above, all optimal controls are continuous, while the case of optimal control with jumps is not considered. Next, we use the interpolation algorithm to solve a control problem in which optimal control has jumps. We attempt to show that the interpolation algorithm is also feasible for this kind of problems.

**Example 5.** Consider the following optimal control problem with two state variables and one control.

$$\left\{ \begin{array}{l} \left( \begin{array}{c} \dot{x}_1(t) \\ \dot{x}_2(t) \end{array} \right) = \left( \begin{array}{c} x_2(t) \\ -x_1(t) + \frac{1}{4}u(t) \end{array} \right), \quad t \in (0, 2\pi], \\ x_1(0) = 0, \quad x_2(0) = 0, \\ J(u(\cdot)) = x_2(2\pi), \\ \min_{(u(\cdot)) \in \mathcal{U}} J(u(\cdot)), \end{array} \right. \quad (3.8)$$

FIGURE 7. Numerical solutions of HJB equation (3.6) in Example 4 at  $x_1 = 0$ .

where  $\mathcal{U} = L^\infty([0, 2\pi]; U)$ ,  $U = [-1, 1]$ . The HJB equation of (3.8) can be written as

$$-\frac{\partial}{\partial t} v(t, x_1, x_2) - \inf_{|u| \leq 1} \left\{ x_2 \cdot v_{x_1}(t, x_1, x_2) + \left(-x_1 + \frac{1}{4}u\right) \cdot v_{x_2}(t, x_1, x_2) \right\} = 0,$$

$$(t, x_1, x_2) \in [0, 2\pi] \times \mathbb{R}^2,$$

$$v(2\pi, x_1, x_2) = x_2, \quad x_2 \in \mathbb{R}.$$

Here we choose the computational region  $(t, x_1, x_2) \in [0, 2\pi] \times \left[-\frac{3}{2}, \frac{3}{2}\right]^2$ . The number of nodes  $N$  on  $t$  axis is chosen as 41. And the number of nodes on  $x_1$  and  $x_2$  axis are set as  $M_1 = M_2 = 11$ . Then the time step  $\Delta t = -\frac{\pi}{20}$  and the space steps  $\Delta x_1 = \Delta x_2 = \frac{3}{10}$ . In this case, for the given initial state  $(x_{10}, x_{20}) = (0, 0)$ , the unique optimal control  $u^*(t)$  and the corresponding optimal trajectory  $(x_1^*(t), x_2^*(t))$

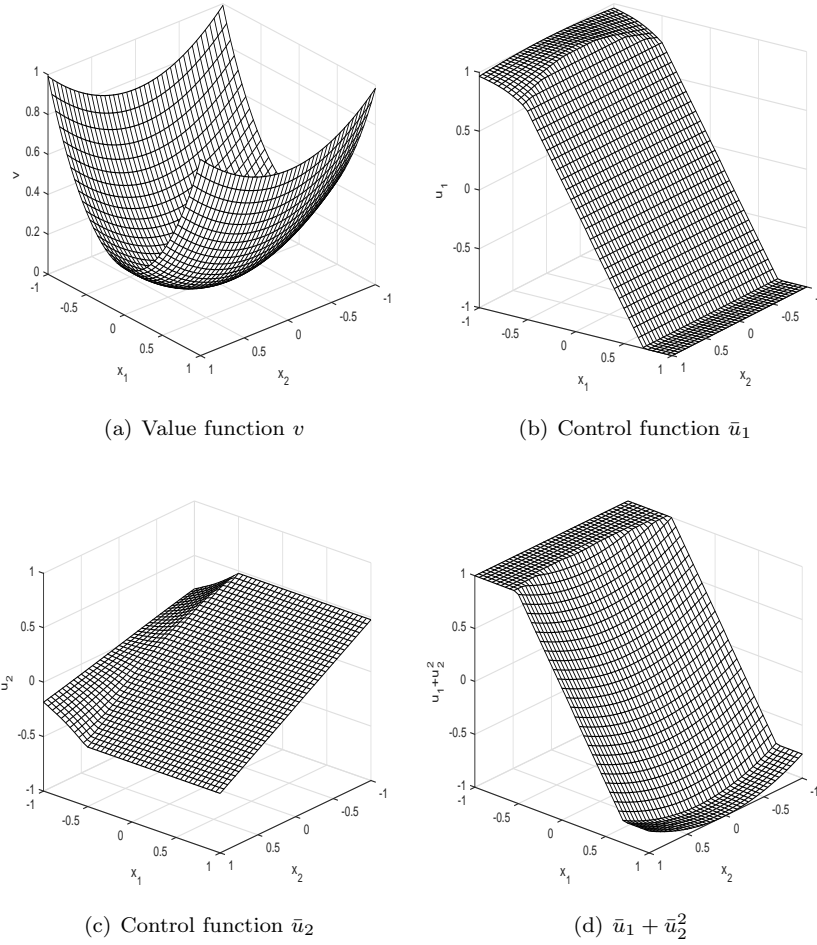


FIGURE 8. Numerical solutions of the HJB equation (3.6) in Example 4 at  $t = \frac{1}{2}$ .

of the system (3.8) can be obtained analytically by

$$u^*(t) = \begin{cases} -1, & 0 \leq t < \frac{\pi}{2}, \\ 1, & \frac{\pi}{2} \leq t < \frac{3\pi}{2}, \\ -1, & \frac{3\pi}{2} \leq t \leq 2\pi, \end{cases} \tag{3.9}$$

and

$$\begin{pmatrix} x_1^*(t) \\ x_2^*(t) \end{pmatrix} = \begin{cases} \frac{1}{4} \begin{pmatrix} \cos t - 1 \\ -\sin t \end{pmatrix}, & 0 \leq t < \frac{\pi}{2}, \\ \frac{1}{4} \begin{pmatrix} \cos t - 2 \sin t + 1 \\ -2 \cos t - \sin t \end{pmatrix}, & \frac{\pi}{2} \leq t < \frac{3\pi}{2}, \\ \frac{1}{4} \begin{pmatrix} \cos t - 4 \sin t - 1 \\ -4 \cos t - \sin t \end{pmatrix}, & \frac{3\pi}{2} \leq t \leq 2\pi. \end{cases}$$



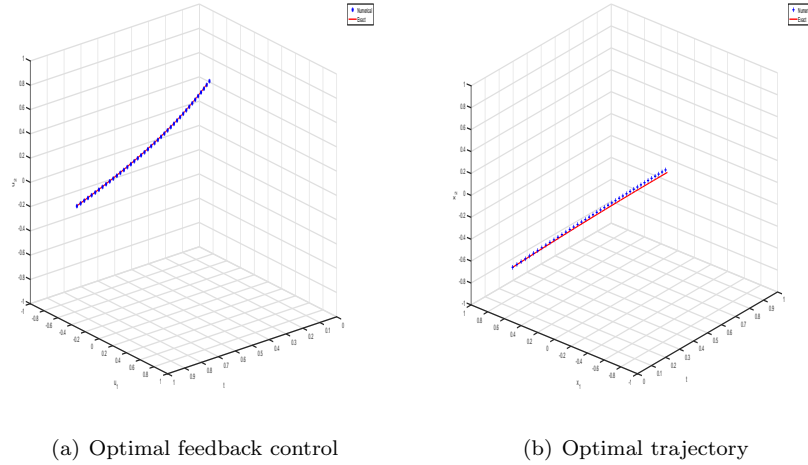


FIGURE 9. Numerical and exact solutions in Example 4 with initial states  $(x_{10}, x_{20}) = (\frac{1}{2}, -\frac{1}{2})$  (Linear interpolation case).

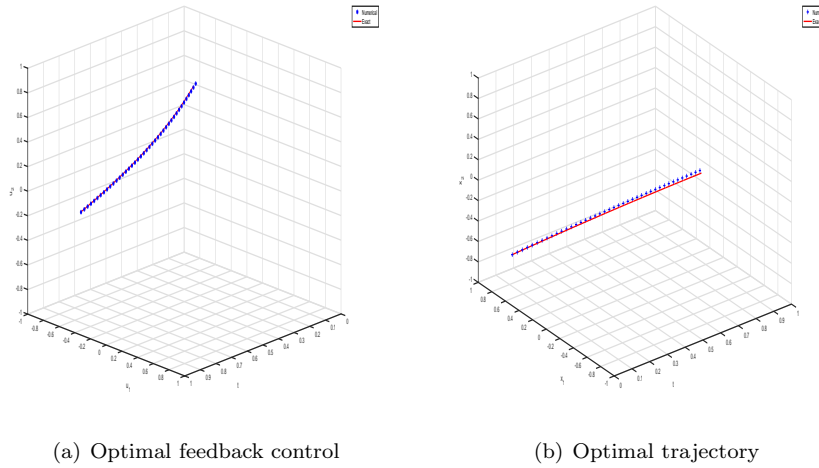


FIGURE 10. Numerical and exact solutions in Example 4 with initial states  $(x_{10}, x_{20}) = (\frac{1}{2}, -\frac{1}{2})$  (Cubic spline interpolation case).

From (3.9), we see that the optimal control is not continuous. It has two jumps in  $[0, 2\pi]$ . For this example, we execute the computation by the interpolation algorithm above. We choose the linear interpolation and the Heun method in Sub-step 3.2. The simulation results are plotted in Figure 11, in which we find all numerical solutions have a good agreement with the exact solutions to optimal control and trajectory. The computational errors in the maximum norms between the numerical solutions and exact solutions to optimal control  $u^*$  and optimal trajectory  $(x_1^*, x_2^*)$  are, respectively,  $4.8000e-05$  and  $0.0911$ . These small error estimates demonstrate that interpolation algorithm is likewise effective for finding the numerical optimal controls for solving control problem with optimal control with jumps.

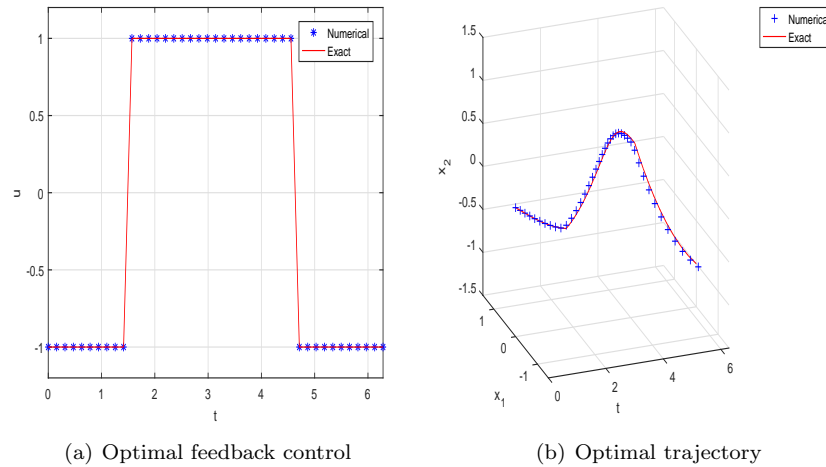


FIGURE 11. Numerical and exact solutions in Example 5 (Linear interpolation case).

**4. Conclusion.** In this paper, we construct an interpolation algorithm for finding an approximation of optimal feedback control. And five experimental examples with different constraints (even nonlinear constraints) on control are successfully solved to demonstrate the effectiveness of the presented interpolation algorithm. The numerical results show that the algorithm with interpolation is very efficient for solving practical optimal control problems.

**Acknowledgments.** The authors would like to thank the editor and the anonymous referees for their very careful reading and constructive suggestions that improve the manuscript substantially.

#### REFERENCES

- [1] A. Alla, *Model Reduction for A Dynamic Programming Approach to Optimal Control Problems with PDE Constraints*, Ph.D thesis, University of Rome in Sapienza, Italy, 2014.
- [2] A. Alla, B. Haasdonk and A. Schmidt, [Feedback control of parametrized PDEs via model order reduction and dynamic programming principle](#), *Adv. Comput. Math.*, **46** (2020), Paper No. 9, 28 pp.
- [3] M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Systems & Control: Foundations & Applications, with appendices by Maurizio Falcone and Pierpaolo Soravia, Birkhäuser, Boston, 1997.
- [4] M. V. Basin and M. A. Pinsky, [Control of Kalman-like filters using impulse and continuous feedback design](#), *Discrete Contin. Dyn. Syst. Ser. B*, **2** (2002), 169–184.
- [5] T. Breiten and K. Kunisch, [Feedback stabilization of the three-dimensional Navier-Stokes equations using generalized Lyapunov equations](#), *Discrete Contin. Dyn. Syst.*, **40** (2020), 4197–4229.
- [6] D. Castorina, A. Cesaroni and L. Rossi, [On a parabolic Hamilton-Jacobi-Bellman equation degenerating at the boundary](#), *Commun. Pure Appl. Anal.*, **15** (2016), 1251–1263.
- [7] M. G. Crandall, Viscosity solutions: A primer, in *Lecture Notes in Mathematics* (eds. I. Capuzzo-Dolcetta and P. L. Lions), Springer-Verlag, Berlin, (1997), 1–43.
- [8] M. G. Crandall, L. C. Evans and P.-L. Lions, [Some properties of viscosity solutions of Hamilton-Jacobi equations](#), *Trans. Amer. Math. Soc.*, **282** (1984), 487–502.
- [9] M. G. Crandall and P.-L. Lions, [Viscosity solutions of Hamilton-Jacobi equations](#), *Trans. Amer. Math. Soc.*, **277** (1983), 1–42.

- [10] G. Fabrini, M. Falcone and S. Volkwein, Coupling MPC and HJB for the computation of POD-based Feedback Laws, *Numerical Mathematics and Advanced Applications—ENUMATH*, (2017), 941–949, Lect. Notes Comput. Sci. Eng., 126, Springer, Cham, 2019.
- [11] M. Falcone and R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*, SIAM, Philadelphia, 2014.
- [12] W. H. Fleming and H. M. Soner, *Controlled Markov Processes and Viscosity Solutions, Stochastic Modelling and Applied Probability*, Springer-Verlag, New York, 2006.
- [13] S. Gombao, Approximation of optimal controls for semilinear parabolic PDE by solving Hamilton-Jacobi-Bellman equations, in *Electronic Proceedings of Fifteenth International Symposium on Mathematical Theory of Networks and Systems* (eds. D. S. Gilliam and J. Rosenthal), South Bend, USA, (2002), 1–15.
- [14] B.-Z. Guo and B. Sun, Dynamic programming approach to the numerical solution of optimal control with paradigm by a mathematical model for drug therapies of HIV/AIDS, *Optim. Eng.*, **15** (2004), 119–136.
- [15] B.-Z. Guo and B. Sun, Numerical solution to the optimal feedback control of continuous casting process, *J. Global Optim.*, **39** (2007), 171–195.
- [16] Y.-C. Ho, On centralized optimal control, *IEEE Trans. Automat. Control*, **50** (2005), 537–538.
- [17] D. Kalise and K. Kunisch, Polynomial approximation of high-dimensional Hamilton-Jacobi-Bellman equations and applications to feedback control of semilinear parabolic PDEs, *SIAM J. Sci. Comput.*, **40** (2018), 629–652.
- [18] K. Kunisch, S. Volkwein and L. Xie, HJB-POD-based feedback design for the optimal control of evolution problems, *SIAM J. Appl. Dyn. Syst.*, **3** (2004), 701–722.
- [19] K. Kunisch and L. Xie, POD-based feedback control of the Burgers equation to solving the evolutionary HJB equation, *Comput. Math. Appl.*, **49** (2005), 1113–1126.
- [20] K. Kunisch and L. Xie, Suboptimal feedback control of flow separation by POD model reduction, in *Real-Time PDE-Constrained Optimization*, (eds. L. T. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes and B. van Bloemen Waanders), Computational Science & Engineering, SIAM, (2007), 233–250.
- [21] H. J. Kushner and P. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time, Stochastic Modelling and Applied Probability*, Springer-Verlag, New York, 2001.
- [22] W. M. McEneaney, Convergence rate for a curse-of-dimensionality-free method for Hamilton-Jacobi-Bellman PDEs represented as maxima of quadratic forms, *SIAM J. Control Optim.*, **48** (2009), 2651–2685.
- [23] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*, Interscience Publishers John Wiley & Sons, Inc., New York, 1962.
- [24] J. E. Rubio, *Control and Optimization: The Linear Treatment of Nonlinear Problems (Nonlinear Science: Theory and Applications)*, Manchester University Press, Manchester, 1986.
- [25] T. Sauer, *Numerical Analysis*, 2<sup>nd</sup> edition, Pearson Education, Essex, 2012.
- [26] C. J. Silva and D. F. M. Torres, A TB-HIV/AIDS coinfection model and optimal control treatment, *Discrete Contin. Dyn. Syst.*, **35** (2015), 4639–4663.
- [27] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Texts in Applied Mathematics, Vol. 12, Springer-Verlag, New York, 2002.
- [28] B. Sun and B.-Z. Guo, Convergence of an upwind finite-difference scheme for Hamilton-Jacobi-Bellman equation in optimal control, *IEEE Trans. Automat. Control*, **60** (2015), 3012–3017.
- [29] S. Wang, F. Gao and K. L. Teo, An upwind finite-difference method for the approximation of viscosity solutions to Hamilton-Jacobi-Bellman equations, *IMA J. Math. Control Inform.*, **17** (2000), 167–178.
- [30] J. M. Yong and X. J. Li, *A Concise Lecture Note on Optimal Control Theory*, Higher Education Press, Beijing, 2006. (in Chinese)
- [31] X. Y. Zhou, Verification theorems within the framework of viscosity solutions, *J. Math. Anal. Appl.*, **177** (1993), 208–225.

Received November 2020; revised May 2021.

E-mail address: 1034635107@qq.com

E-mail address: sun345@bit.edu.cn