

# Solving Linear Optimal Control Problems Using Cubic B-spline Quasi-interpolation

M. Matinfar\* and M. Dosti

University of Mazandaran

Department of Mathematics, Science of Mathematics Faculty, Babolsar, Iran.

\*Corresponding author: m.matinfar@umz.ac.ir

Article history

Received: 11 April 2016

Received in revised form: 17 February 2018

Accepted: 2 April 2018

Published on line: 1 December 2018

---

**Abstract** In this article, we apply method for solving linear optimal control problem based on cubic B-spline quasi-interpolation. Hamilton-Jacobi equation are applied to linear optimal control problem can be converted to systems of first-order equations. The main idea of our scheme is approximation derivative with cubic B-spline quasi-interpolation. This method is straightforward, without restrictive assumptions. The results of scheme are made in pleasant agreement with analytic solutions. The accuracy of the proposed method is demonstrated by absolute error. Our scheme is simple to implement because its algorithm is easy and it's one of the advantages of the proposed method.

**Keywords** Cubic B-spline; quasi-interpolation; optimal control; Hamilton

**Mathematics Subject Classification** 65K10, 49M05, 93B40.

## 1 Introduction

Linear optimal control problem has the following form:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \quad x(t_0) = x_0, \\ J &= \frac{1}{2}x^T(t_f)Sx(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (x^T Qx + u^T Ru)dt, \end{aligned} \quad (1)$$

where  $u \in \mathbf{R}^m$ ,  $x \in \mathbf{R}^n$ ,  $A \in \mathbf{R}^{n \times n}$  and  $B \in \mathbf{R}^{n \times m}$ . If the control  $u(t)$  is piecewise continuous in  $t$  for each  $t$ , it's an admissible control and its values belong to a given closed subset  $U$  of  $\mathbf{R}^+$ . The input  $u(t)$  is derived by minimizing the quadratic performance index  $J$ , where  $S$  and  $Q$  are symmetric positive semi-definite and  $R$  is symmetric positive definite. The optimal control law,  $u(t) = k(t)x(t)$ , are achieved for system (1) by using Pontryagin's maximum principle, [1].

Optimal control problems (OCP) arise in many applications such as economics, chemical engineering, robotics, aeronautics. In some cases, it is not easy to obtain analytical solution of generally OCP's therefore numerical methods are acceptable way to solve many of these

problems. Yousefi et al. studied He's variational iteration method for linear optimal control, [1]. In [2], Optimal control is solved by homotopy perturbation method.

Pseudospectral collocation methods for the direct transcription of OCP's was presented by Pietz, [3]. In [4] the OCP is solved by using state parametrization.

Since in [1] the error of method is not explicit, we will report error of method as table and graphically.

We try to develop numerical method that is accurate and efficient. In this work, we obtain an approximate value for  $k(t)$  by cubic B-spline quasi-interpolation (BSQI). Quasi-interpolation is very useful in the study of approximation theory and its applications. BSQI scheme are applied in diffrenet partial differential equations, [8]- [9]. Zhu and Wang used this method for numerical solution Burgers equation, [10] and Zhu and Kang solved Burgers-Fisher equation based on BSQI, [11]. Various degrees B-spline are used to solve many ordinary differential equations or partial differential equations and quasi-interpolation are studied for the numerical solutions of differential equations, [5]- [7]. We compute the B-spline coefficients of the approximation solution and assume that the knot and degree of B-spline are known.

The layout of this paper is as follows: First we introduce cubic B-spline quasi-interpolation and then present the formulation of our method in section 2. Computational results of problem are illustrated and compared with analytical solution in section 3. We end this work with conclusion and references.

## 2 Method of Solution

In this section, first we represent B-spline quasi-interpolants briefly and then the numerical scheme is presented for linear optimal control problems based on the techniques using matrix arguments and cubic B-spline quasi-interpolation.

### 2.1 Univariate B-spline Quasi-interpolants

We assume uniform partition  $T_f = \{t_i = a + ih, i = 0, \dots, f\}$  with the steplength  $h = (t_f - t_0)/f$  for  $I = [a, b]$ , where  $a = t_0$  and  $b = t_f$ . Let  $S_d(X_f)$  be the space of univariate splines of degree  $d$  and  $C^{d-1}$  on this partition. Suppose  $\{B_j; j \in J\}$  with  $J = \{1, 2, \dots, f + d\}$  is the B-spline basis of  $S_d(X_f)$ , which can be computed by the de Boor-Cox formula, [12]. With these notations, the support of  $B_j$  is  $supp(B_j) = [t_{j-d-1}, t_j]$ . As usual, we add multiple knots at the endpoints:  $a = t_{-d} = t_{-d+1} = \dots = t_0$  and  $b = t_f = t_{f+1} = \dots = t_{f+d}$ .

In [13], univariate B-spline quasi-interpolants can be defined as operators of the form

$$Q_d g = \sum_{j \in J} \mu_j(g) B_j, \quad (2)$$

where  $\mu_j$  is a linear combination of discrete values of  $g$  at some points in the neighborhood of  $supp(B_j)$ .

According to [13] the space of polynomials of total degree at most  $d$  are denoted by  $\prod_d$ . In general we impose that  $Q$  is exact on the space  $\prod_d$ , i.e.  $Q_p = p$  for all  $p \in \prod_d$ .

Cubic B-spline basis  $B_j$  can be computed when we use the de Boor-Cox formula [12] and [13] for  $j \in J$ . Let  $g_i = g(t_i)$ ,  $i = 0, 1, \dots, f$ . A quasi-interpolant based on cubic B-splines is given

by

$$Q_3g = \sum_{j=1}^{f+3} \mu_j(g)B_j, \tag{3}$$

where the coefficients are listed as follows:

$$\begin{cases} \mu_1(g) = g_0, \\ \mu_2(g) = \frac{1}{18}(7g_0 + 18g_1 - 9g_2 + 2g_3), \\ \mu_j(g) = \frac{1}{6}(-g_{j-3} + 8g_{j-2} - g_{j-1}), \quad j = 3, \dots, f + 1, \\ \mu_{f+2}(g) = \frac{1}{18}(2g_{f-3} - 9g_{f-2} + 18g_{f-1} + 7g_f), \\ \mu_{f+3}(g) = g_f. \end{cases} \tag{4}$$

For  $g \in C^4(I)$ , we have the error estimate [13]

$$\|g - Q_3g\|_{\infty, I_k} \leq \frac{8}{3}d_{\infty, I_k}(g, \Pi_3) \quad \text{for } 1 \leq k \leq n \Rightarrow \|g - Q_3g\|_{\infty} = O(h^4). \tag{5}$$

Differentiating interpolation polynomials leads to classical finite differences for the approximate computation of derivatives. Therefore, it seems natural to approximate derivatives of  $g$  by derivatives of  $Q_3g(x)$  up to the order  $h^3$ . We can evaluate the value of  $g$  at  $x_i$  by  $(Q_3g)' = \sum_{j=1}^{n+3} \mu_j(g)B'_j$ . For  $j \in J$ ,  $B'_j$  are computed and then we have

$$\begin{aligned} Qf'(t_0) &= \frac{1}{h}\left(-\frac{11}{6}g_0 + 3g_1 - \frac{3}{2}g_2 + \frac{1}{3}g_3\right), \\ Qg'(t_1) &= \frac{1}{h}\left(-\frac{1}{3}g_0 - \frac{1}{2}g_1 + g_2 - \frac{1}{6}g_3\right), \\ Qg'(t_j) &= \frac{1}{h}\left(\frac{1}{12}g_{j-2} - \frac{2}{3}g_{j-1} + \frac{2}{3}g_{j+1} - \frac{1}{12}g_{j+2}\right), \quad 2 \leq j \leq f - 2, \\ Qg'(t_{f-1}) &= \frac{1}{h}\left(\frac{1}{6}g_{f-3} - g_{f-2} + \frac{1}{2}g_{f-1} + \frac{1}{3}g_f\right), \\ Qg'(t_f) &= \frac{1}{h}\left(-\frac{1}{3}g_{f-3} + \frac{3}{2}g_{f-2} - 3g_{f-1} + \frac{11}{6}g_f\right). \end{aligned} \tag{6}$$

### 2.2 Solution of the Linear Optimal Control System

Hamiltonian for Eq.(1) are given as

$$H(x, u, \lambda, t) = \frac{1}{2}(x^T Qx + u^T Ru) + \lambda^T(Ax + Bu), \tag{7}$$

where  $\lambda \in \mathbf{R}^n$  is known as the costate variable. The optimal control by Pontryagin’s maximum principle is obtained by solving

$$\frac{\partial H}{\partial u} = Ru + B^T \lambda = 0, \tag{8}$$

where  $\lambda$  is a solution of the adjoint equation

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = -Qx - A^T \lambda, \quad (9)$$

with the terminal condition

$$\lambda(t_f) = Sx(t_f). \quad (10)$$

Thus, it follows from Eq. (8) that the optimal control law is

$$u^*(t) = -R^{-1}B^T \lambda(t), \quad (11)$$

where  $\lambda(t)$  is the solution of the Hamiltonian system

$$\begin{pmatrix} \dot{x} \\ \dot{\lambda} \end{pmatrix} = \begin{pmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad (12)$$

which follows from Eq. (1) with  $u$  substituted from Eq. (9) and Eq. (11). The optimal control can be implemented as a closed loop if the solution to the adjoint Eq. (9) is assumed like Eq. (10) as a linear function of the states in the form

$$\lambda(t) = p(t)x(t), \quad p(t_f) = S. \quad (13)$$

By using Eqs. (9),(12) and (13), we have

$$\begin{aligned} \dot{x}(t) &= Ax(t) - BR^{-1}B^T p(t)x(t), \\ \dot{\lambda}(t) &= \dot{p}(t)x(t) + p(t)\dot{x}(t) \\ &= [\dot{p}(t) + p(t)A - p(t)BR^{-1}B^T p(t)]x(t) \\ &= [-Q - A^T p(t)]x(t), \end{aligned}$$

where the first equality follows from Eq. (13) and the second one from Eq. (12). Hence,

$$[\dot{p}(t) + p(t)A + A^T p(t) + Q - p(t)BR^{-1}B^T p(t)]x(t) = 0, \quad (14)$$

Since the above equation must hold for all nonzero  $x(t)$ ,  $p(t)$  must satisfy the matrix Riccati equation,

$$-\dot{p}(t) = p(t)A + A^T p(t) + Q - p(t)BR^{-1}B^T p(t), \quad p(t_f) = S. \quad (15)$$

By considering Eqs.(11) and (13), we can see that the optimal control law is given by [1]

$$u^*(t) = -R^{-1}B^T p(t)x(t), \quad (16)$$

and as shown in [14],  $p(t)$  can be computed using the following relation

$$p(t) = W(t)V^{-1}(t), \quad (17)$$

where

$$\begin{pmatrix} \dot{V}(t) \\ \dot{W}(t) \end{pmatrix} = \begin{pmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{pmatrix} \begin{pmatrix} V(t) \\ W(t) \end{pmatrix}, \quad (18)$$

with conditions,  $V(t_f) = I$  and  $W(t_f) = S$ .

This system of equations involving the first-order derivatives can be computed by employing the cubic B-spline quasi-interpolation.

Now we get approximate solution of linear systems of equations as follows:

$$V(t) = Q_3 V(t) = \sum_{m=1}^{N+3} \alpha_m(V) B_m(t), \tag{19}$$

$$W(t) = Q_3 W(t) = \sum_{m=1}^{N+3} \beta_m(W) B_m(t), \tag{20}$$

Obviously, derivative of the cubic B-spline quasi-interpolant  $Q_3 \dot{V}(t)$  and  $Q_3 \dot{W}(t)$  are used to approximate  $\dot{V}(t)$  and  $\dot{W}(t)$

$$Q_3 \dot{V}(t) = \sum_{m=1}^{N+3} \alpha_m(V) B'_m(t), \tag{21}$$

$$Q_3 \dot{W}(t) = \sum_{m=1}^{N+3} \beta_m(W) B'_m(t), \tag{22}$$

We assume system (18) at time  $t_i, i = 0, 1, \dots, f$ ,

$$\begin{cases} \dot{V}(t_i) = AV(t_i) - BR^{-1}B^TW(t_i), \\ \dot{W}(t_i) = -QV(t_i) - A^TW(t_i). \end{cases} \tag{23}$$

Substituting Eqs. (21)-(22) into Eq.(18) then is obtained as follows

$$\sum_{m=1}^{N+3} \alpha_m(V) B'_m(t_i) = AV(t_i) - BR^{-1}B^TW(t_i), \tag{24}$$

$$\sum_{m=1}^{N+3} \beta_m(W) B'_m(t_i) = -QV(t_i) - A^TW(t_i), \tag{25}$$

and conditions

$$\sum_{m=1}^{N+3} \alpha_m(V) B_m(t_f) = I, \sum_{m=1}^{N+3} \beta_m(W) B_m(t_f) = S. \tag{26}$$

We continue with the discretization of the system of Eqs. (24)-(25) and the value of B-spline functions are determined using Eqs. (6) at the knots  $t_i, i = 0, 1, \dots, f$  . We obtain

$$\begin{aligned} t &= t_0, \\ \left(\frac{-11}{6h} - A\right)V_0 + \frac{3}{h}V_1 - \frac{3}{2h}V_2 + \frac{1}{3h}V_3 + BRB^{-1}W_0 &= 0, \\ QV_0 + \left(\frac{-11}{6h} + A^T\right)W_0 + \frac{3}{h}W_1 - \frac{3}{2h}W_2 + \frac{1}{3h}W_3 &= 0, \end{aligned}$$

$$\begin{aligned}
 &t = t_1, \\
 &-\frac{1}{3h}V_0 - (A + \frac{1}{2h})V_1 + \frac{1}{h}V_2 - \frac{1}{6h}V_3 + BRB^{-1}W_1 = 0, \\
 &QV_1 - \frac{1}{3h}W_0 + (A^T - \frac{1}{2h})W_1 + \frac{1}{h}W_2 - \frac{1}{6h}W_3 = 0, \\
 &t = t_j, \quad j = 2, \dots, f - 2, \\
 &\frac{1}{12h}V_{j-2} - \frac{2}{3h}V_{j-1} - AV_j + \frac{2}{3h}V_{j+1} - \frac{1}{12h}V_{j+2} + BRB^{-1}W_j = 0, \\
 &QV_j + \frac{1}{12h}W_{j-2} - \frac{2}{3h}W_{j-1} + A^TW_j + \frac{2}{3h}W_{j+1} - \frac{1}{12h}W_{j+2} = 0, \\
 &t = t_{f-1}, \\
 &\frac{1}{6h}V_{f-3} - \frac{1}{h}V_{f-2} - (A - \frac{1}{2h})V_{f-1} + \frac{1}{3h}V_f + BRB^{-1}W_{f-1} = 0, \\
 &QV_{f-1} + \frac{1}{6h}W_{f-3} - \frac{1}{h}W_{f-2} + (A^T - \frac{1}{2h})W_{f-1} + \frac{1}{3h}W_f = 0, \\
 &t = t_f, \\
 &\frac{-1}{3h}V_{f-3} + \frac{3}{2h}V_{f-2} - \frac{3}{h}V_{f-1} + (-A + \frac{11}{6})V_f + BRB^{-1}W_f = 0, \\
 &QV_f - \frac{1}{3}W_{f-3} + \frac{3}{2h}W_{f-2} - \frac{3}{h}W_{f-1} + (\frac{11}{6h} + A^T)W_f = 0.
 \end{aligned} \tag{27}$$

### 3 Computational Results

In this section, we will use BSQI scheme to obtain approximate solution of  $k(t)$ . Also, we have,  $k(t) = R^{-1}B^T p(t)$  and  $u^*(t) = -k(t)x(t)$ . This method is tested on three examples and absolute errors are calculated to represent that the proposed method is viable and efficient.

**Example 1:** Consider system (1) as follows:

$$\begin{aligned}
 \dot{x}(t) &= -2x(t) + u(t), \\
 J &= \frac{1}{2}x^2(1) + \frac{1}{2} \int_0^1 [x^2(t) + u^2(t)] dt.
 \end{aligned}$$

where  $R = 1, B = 1, A = -2, Q = 1, S = 1$  and  $t_f = 1$ . This system is a single-input scalar system.

Approximate and exact curves of  $k(t)$  is shown for  $h = 0.0005$  in Figure 1. The analytical solution of this example is

$$k(t) = \frac{\lambda \cosh \lambda(1 - t) - \sinh \lambda(1 - t)}{\lambda \cosh \lambda(1 - t) + 3 \sinh \lambda(1 - t)}, \quad \text{where } \lambda = \sqrt{5}.$$

Accuracy of algorithms is shown by calculating the absolute error. The results are shown in Table 1. Absolute error are graphed in Figure 2.

**Example 2:** Give  $R = 1, B = 1, A = -1, Q = 1, S = 0$  and  $t_f = 1$ . We have system (1) as

$$\begin{aligned}
 \dot{x}(t) &= -x(t) + u(t), \\
 J &= \frac{1}{2} \int_0^1 [x^2(t) + u^2(t)] dt.
 \end{aligned}$$

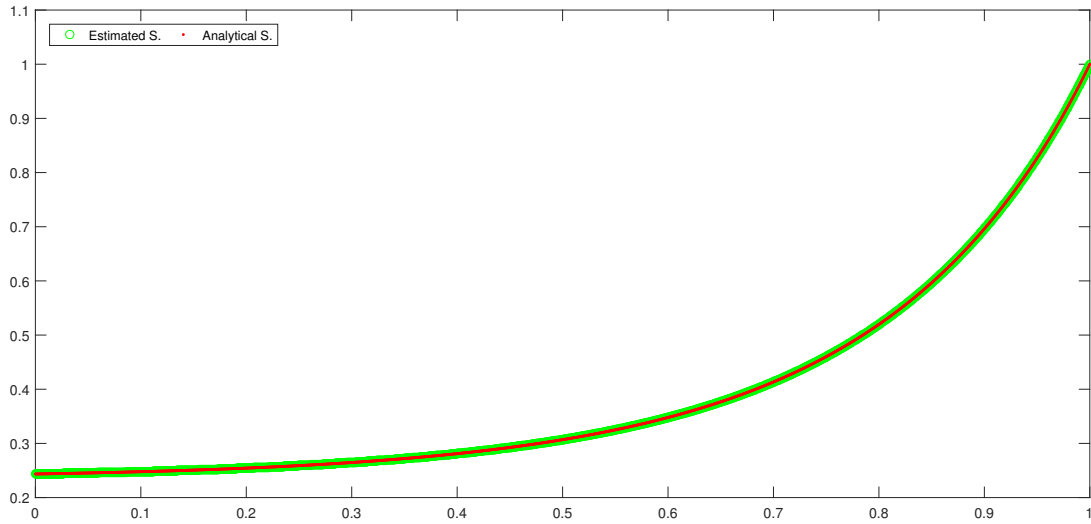


Figure 1: Comparison between the Estimated Solution with Analytical Solution of Example 1.

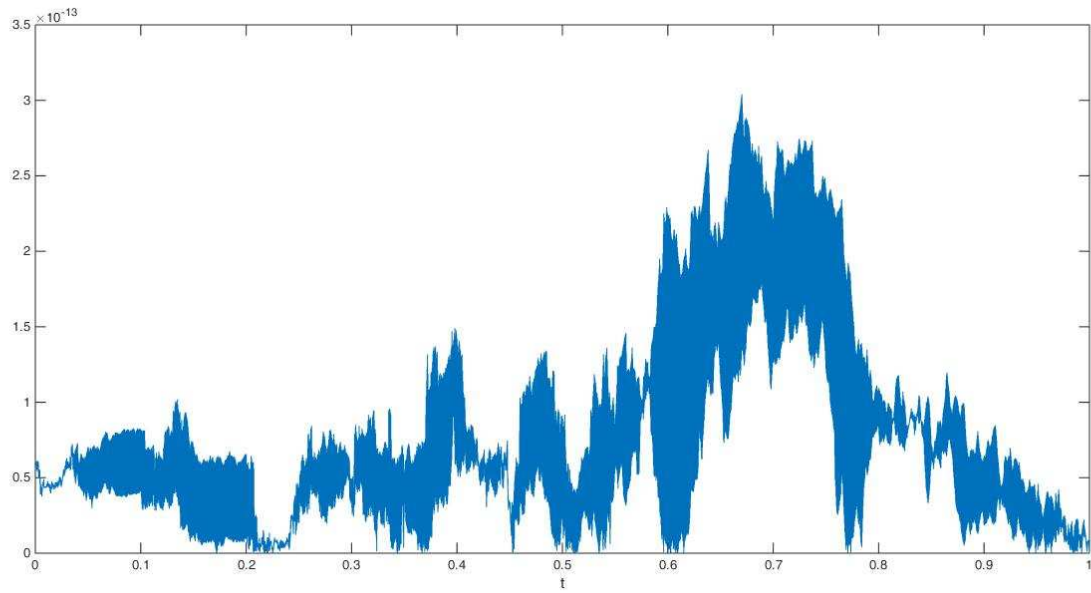


Figure 2: Absolute Error of Solution for Example 1 with  $h = 0.0002$

Table 1: Absolute Error between Exact and Estimate Solution

	Exact S.	Estimate S. $h = 0.0002$	Absolute E.	Estimate S. $h = 0.002$	Absolute E.	Estimate S.	Absolute E. $h = 0.02$
$t = 0$	0.2435	0.2435	$6.1978 \times 10^{-14}$	0.2435	$4.4292 \times 10^{-13}$	0.2435	$4.2493 \times 10^{-9}$
$t = 0.2$	0.2543	1.2604	$6.5225 \times 10^{-14}$	0.2543	$7.9430 \times 10^{-13}$	0.2543	$7.8952 \times 10^{-9}$
$t = 0.4$	0.2811	0.2811	$1.4194 \times 10^{-13}$	0.2811	$1.5344 \times 10^{-12}$	0.2811	$1.4823 \times 10^{-8}$
$t = 0.6$	0.3478	0.3478	$9.9920 \times 10^{-15}$	0.3478	$2.9366 \times 10^{-12}$	0.3478	$2.7644 \times 10^{-8}$
$t = 0.8$	0.5197	0.5197	$8.2601 \times 10^{-14}$	0.5197	$4.7896 \times 10^{-12}$	0.5197	$4.1491 \times 10^{-8}$
$t = 1$	1	1.0000	$2.2204 \times 10^{-16}$	1.0000	$6.6613 \times 10^{-16}$	1.0000	$1.0109 \times 10^{-9}$

The analytical solution of a single-input scalar system with above assumptions is

$$k(t) = -\frac{(1 + \sqrt{2}\alpha) \cosh(\sqrt{2}t) + (\alpha + \sqrt{2}) \sinh(\sqrt{2}t)}{\cosh(\sqrt{2}t) + \alpha \sinh(\sqrt{2}t)},$$

where  $\beta = -\frac{\cosh(\sqrt{2}) + \sqrt{2} \sinh(\sqrt{2})}{\sqrt{2} \cosh(\sqrt{2}) + \sinh(\sqrt{2})}$ .

We apply Eqs. (28) for this system then we calculate absolute error and document in Table 2 and show graph of error in Figure 3. Comparing between approximate solution and analytical solution are plotted in Figure 4.

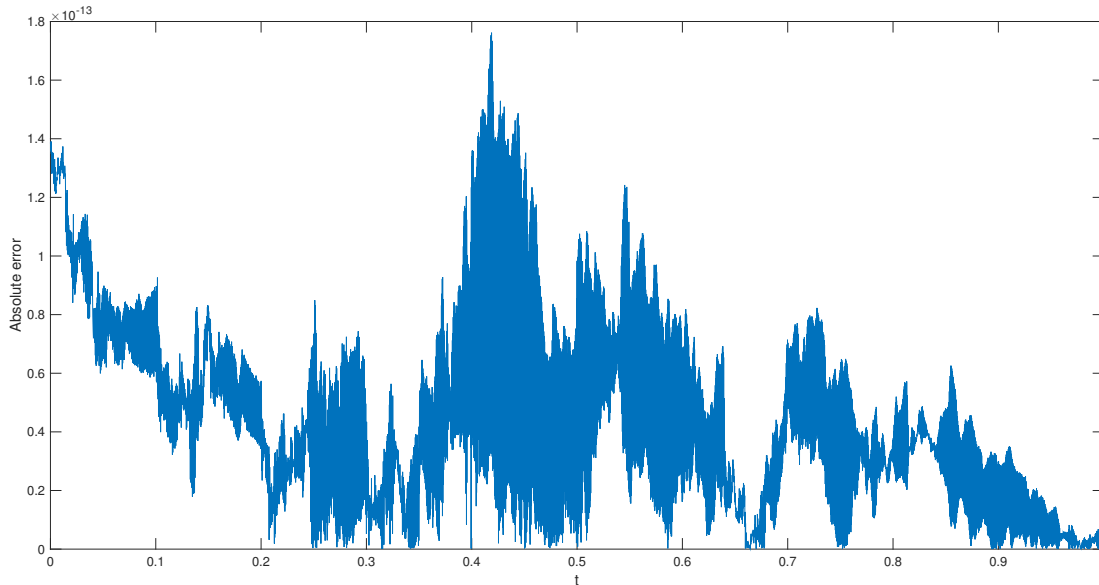


Figure 3: Absolute Error of Solution for Example 2 with  $h = 0.0002$

**Example 3:** A single-input second-order system described by:

$$\dot{x}(t) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t), \tag{28}$$

$$J = \frac{1}{2} \int_0^{\pi/2} [x^T(t) \begin{pmatrix} 0 & 0 \\ 0 & 4 \end{pmatrix} x(t) + u^2(t)] dt. \tag{29}$$



Table 2: Absolute Error between Exact and Estimated Solution for Example 2

	<b>Exact S.</b>	<b>Absolute E. <math>h = 0.0002</math></b>	<b>Absolute E. <math>h = 0.002</math></b>	<b>Absolute E. <math>h = 0.02</math></b>
$t = 0$	0.3858	$1.3855 \times 10^{-13}$	$1.8013 \times 10^{-13}$	$1.6078 \times 10^{-9}$
$t = 0.2$	0.3646	$5.7398 \times 10^{-14}$	$2.3969 \times 10^{-13}$	$2.2222 \times 10^{-9}$
$t = 0.4$	0.3281	$1.3389 \times 10^{-13}$	$2.9004 \times 10^{-13}$	$2.8394 \times 10^{-9}$
$t = 0.6$	0.2659	$6.5392 \times 10^{-14}$	$3.6254 \times 10^{-13}$	$3.2641 \times 10^{-9}$
$t = 0.8$	0.1631	$2.9531 \times 10^{-14}$	$3.1780 \times 10^{-13}$	$2.6559 \times 10^{-9}$
$t = 1$	$1.1812 \times 10^{-15}$	$1.1876 \times 10^{-15}$	$1.2191 \times 10^{-15}$	$1.1851 \times 10^{-12}$

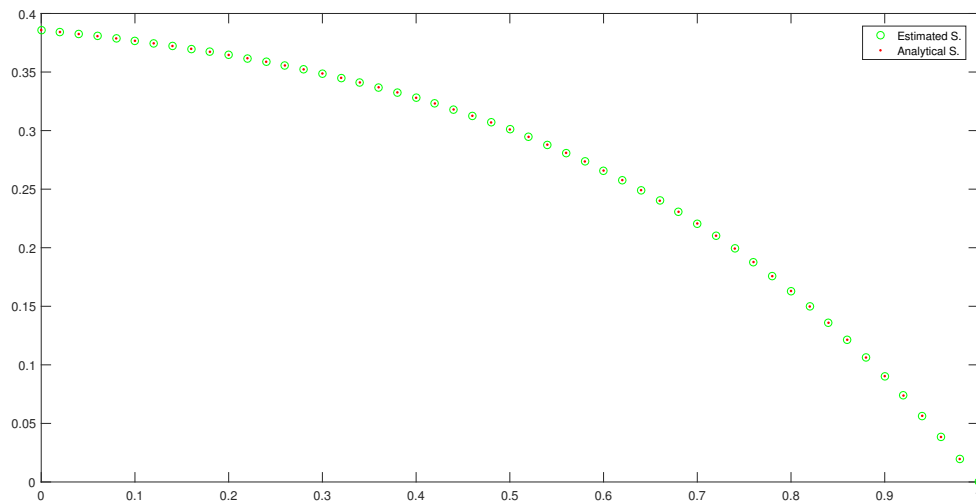


Figure 4: Comparison between the Estimated Solution with Analytical Solution with  $h = 0.02$  for Example 2

where  $t_f = \pi/2$ ,  $S = 0$ ,  $R = 1$ ,  $A = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$ ,  $B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ , and  $Q = \begin{pmatrix} 0 & 0 \\ 0 & 4 \end{pmatrix}$ .

The analytical  $k(t) = [k_1(t), k_2(t)]$  is as follows:

$$k_1(t) = \frac{\sinh(\pi - 2t) - \sin(\pi - 2t)}{\cosh^2(\pi - 2t)/2 + \cos^2(\pi - 2t)},$$

$$k_2(t) = \frac{\cosh(\pi - 2t) - \cos(\pi - 2t)}{\cosh^2(\pi - 2t)/2 + \cos^2(\pi - 2t)}.$$

Table 3 shows the absolute error between approximation and exact values. Also error is presented in Figure 5. The both curves solution are displayed in Figure 6.

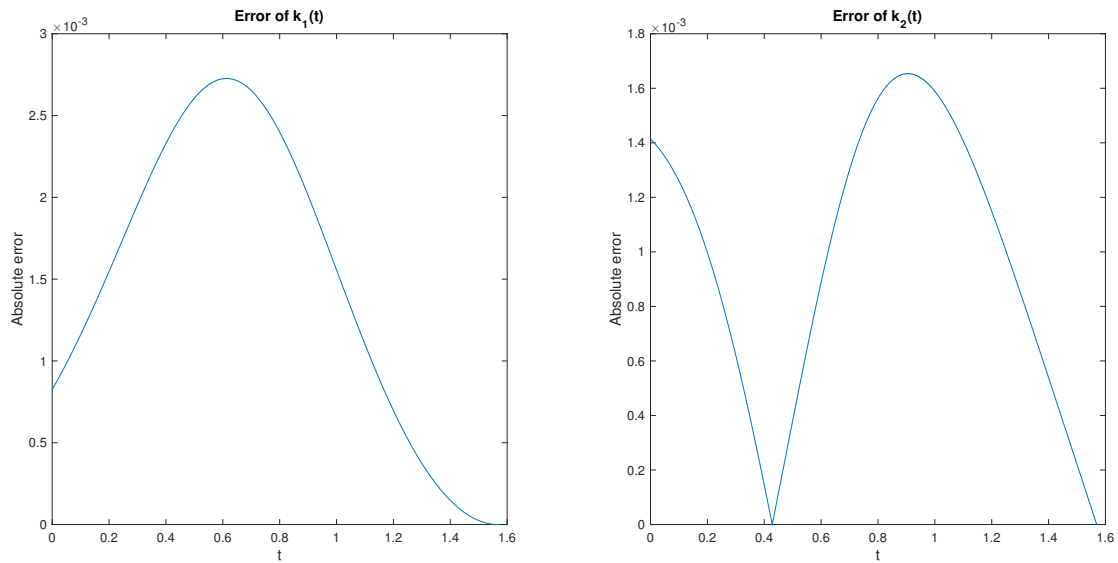


Figure 5: Absolute Error of Solution for Example 3 with  $h = 0.001$

Table 3: Absolute Error between Exact and Estimate solution for Example 3

	Exact Solution for $k_1(t)$	Absolute E. for $k_1(t)$ in $h = 0.001$	Absolute E. for $k_1(t)$ in $h = 0.01$	Exact Solution for $k_2(t)$	Absolute E. for $k_2(t)$ in $h = 0.001$	Absolute E. for $k_2(t)$ in $h = 0.01$
$t = 0$	1.8335	$8.2573 \times 10^{-4}$	$3.7508 \times 10^{-3}$	2	$1.4144 \times 10^{-3}$	$1.4070 \times 10^{-2}$
$t = 0.3$	1.5173	$1.9543 \times 10^{-3}$	$8.6645 \times 10^{-3}$	1.9076	$6.2050 \times 10^{-4}$	$1.1379 \times 10^{-2}$
$t = 0.6$	0.9527	$2.7244 \times 10^{-3}$	$1.1888 \times 10^{-2}$	1.5090	$8.8816 \times 10^{-4}$	$4.7610 \times 10^{-3}$
$t = 0.9$	0.3764	$2.0111 \times 10^{-3}$	$8.6259 \times 10^{-3}$	0.8505	$1.6537 \times 10^{-3}$	$2.0520 \times 10^{-4}$
$t = 1.2$	0.0669	$6.9720 \times 10^{-4}$	$2.8934 \times 10^{-3}$	0.2735	$1.1497 \times 10^{-3}$	$8.2228 \times 10^{-4}$
$t \approx \pi/2$	$5.0121 \times 10^{-7}$	$5.0053 \times 10^{-7}$	$5.6652 \times 10^{-5}$	$1.2682 \times 10^{-6}$	$8.1132 \times 10^{-7}$	$5.7664 \times 10^{-5}$

We see in our examples that the error decreased when the number of points partition are increased or the steplength is chosen small. In other words, the steplength has direct ratio with

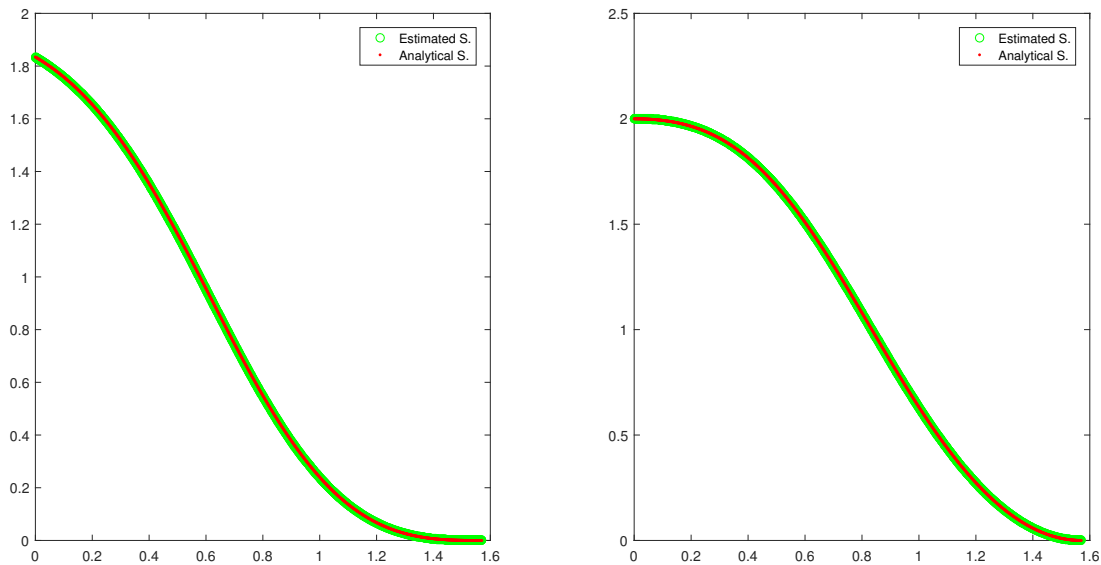


Figure 6: Comparison between the Estimated Solution with Analytical Solution with  $h = 0.001$  for Example 3

error. In single-input scalar system, error is better than single-input second-order system. Also end points had good agreement with analytical solution.

## 4 Conclusion

In this study, cubic B-spline quasi-interpolation was applied for finding the numerical solutions for the linear Optimal Control Problems. Our scheme is feasible and we can see that the error is acceptable in the numerical result. The implementation of the present method is easy, credible and reliable. This method thus eliminates the difficulties and massive computational work. The authors are very grateful to the anonymous referees and editor for their comments.

## References

- [1] Datta, K. B., and Mohan, B.M. *Orthogonal Functions in Systems and Control*. Singapore, New Jersey, London, Hong Kong: World Scientific. 1995.
- [2] Yousefi, S. A., Dehghan, M., and Lotfi, A. Finding the optimal control of linear system via He's variational iteration method. *International journal of Computer Mathematics*. 2010. 87(5): 1042-1050.
- [3] Pietz. J. A. *Estimating the survival functions for right-censored and interval-censored data with piecewise constant hazard functions*. Rice University: Master Thesis. 2003.
- [4] Hashemi Mehne, H., and Hashemi Borzabadi, A. A numerical method for solving optimal control problem using state parametrization. *Numerical Algorithms*. 2006. 42(2): 165–169.

- [5] Kumar, R., and Baskar, S. B-spline quasi-interpolation based numerical methods for some Sobolev type equations *Journal of Computational and Applied Mathematics*. 2016. 292: 41–66.
- [6] Aminikhah, H., and Alavi, J. Numerical solution of convection diffusion equation using cubic B-spline quasi-interpolation. *Thai Journal of Mathematics*. 2015. 1;14(3):599–613.
- [7] Lakestani, M., and Dehghan, M. Numerical solution of Riccati equation using the cubic B-spline scaling functions and Chebyshev cardinal functions. *Computer Physics Communications*. 2010. 181: 957–966.
- [8] Aminikhah, H., and Alavi, J. Numerical study of the nonlinear Cauchy diffusion problem and Newell-Whitehead equation via cubic B-spline quasi-interpolation. *Iranian Journal of Numerical Analysis and Optimization*. 2015. 5(1): 63–72.
- [9] Li, M., Chen, L., and Ma, Q. A meshfree quasi-interpolation method for solving Burgers equation. In *Mathematical Problems in Engineering*, 2014. 2014.
- [10] Zhu, C. G., and Wang, R. H. Numerical solution of Burgers equation by cubic B-spline quasi-interpolation. *Applied Mathematics and Computation*. 2009. 208: 260–272.
- [11] Zhu, C. G., and Kang, W. S. Numerical solution of Burgers-Fisher equation by cubic B-spline quasi-interpolation. *Applied Mathematics and Computation*. 2010. 216: 2679–2686.
- [12] Farin, G. *Curves and Surfaces for CAGD*, San Francisco: Morgan Kaufman. 2001.
- [13] Sablonnière, P. Univariate spline quasi-interpolants and applications to numerical analysis. *Rend. Sem. Mat. Univ. Pol. Torino*. 2005. 63: 211–223.
- [14] Ghorbani, A., and Saberi-Nadjafi, J. An effective modification of He’s variational iteration method. *Nonlinear Anal. RealWorld Appl.*. 2009. 10: 2828–2833.