Eng. & Tech. Journal, Vol.28, No.18, 2010

A New Computational Method for Optimal Control Problem with B-spline Polynomials

Dr. Suha N. Al-Rawi *, Fuad A. Al-Heety* & Saba S. Hasan*

Received on: 13/ 4/ 2010
Accepted on: 5/ 8/2010

Abstract

The main purpose of this work is to propose direct method which is employed by using state vector parameterization (SVP) to convert the quadratic optimal control problems into quadratic programming problem. The state vector parameterization is based on the spline polynomial which includes: B-spline as a basis functions to approximate the system state variables by a finite length of the basis functions series of unknown parameters. An example as application of this method is given

Keywords: Optimal control, direct method, B-spline polynomials

طريقة حسابية جديدة لحل مسألة السيطرة المثلى باستخدام متعددة حدود الثلمة والمتضمنة متعددات الثلمة التوصيلية

الخلاصة

الغرض الاساسي لهذا العمل هوعرض طريقة مباشرةوالتي وظفت باستخدام أسلوب معلمات متجة الحالة لتحويل مسائل السيطرة المثلى التربيعية الى مسألة البرمجة التربيعية.معلمات متجه الحالة أسست على متعددة حدود الثلمة والمتضمنة متعددات الثلمة التوصيلية كدوال اساسية لتقريب منظومة متغيرات الحالة بواسطة طول محدد من سلسلة الدوال الأساسية بالمعلمات المجهولة.تم اعطاء مثال كتطبيق لهذه الطريقة..

1-Introduction:[7]

Optimal control is a special type of the optimization problem and has tremendous application. It deals with the study of systems. The early developments of the theories were given by engineers whose systems were machines and their interactions were controls. In order to understand a system, mathematical equation representing it exactly or to reasonable approximation is written. Usually a system is represented by one of the equations: differential, partial-differential, integral, Integro-differential, difference, stochastic-differential and stochastic–integral equations, and such equations are known as models of the system.Till around 1950 mathematicians and physicists were by and large engaged in finding the mathematical models of various systems. The absence of physical laws in certain case created problems. However, these were overcome by knowledge of inputs and outputs of system or just statistical data.The side-by side existence and uniqueness of solution was studied which helped to understand the system properly. Since 1948, the scientists have started identifying the factors affecting the behavior of the system. Year's later serious thinking started as

*Applied Science Department, University of Technology/ Baghdad

https://doi.org/10.30684/etj.28.18.8
2412-0758/University of Technology-Iraq, Baghdad, Iraq

to how these factors could be controlled to give the best possible result. This concept has been pursued in a systematic manner in the last 30 years and nowadays is know as optimal control.

The system objective is a given state or set of state which may vary with time .Restrictions or constraints, as they are normally called ,are placed on the set of controls (inputs) to the system ; controls satisfying these constraints belong to the set of admissible control

1-Formal Problem Statement

The optimal control problems considered in this work are defined in terms of the system dynamics, the boundary conditions, and the cost criterium

- System dynamics: the system dynamics will be defined in terms of state space equations

$$\dot{x}_k = f_k(t, x_1, x_2, ..., x_n, u_1, u_2, ..., u_m)$$

;k=1,2,…,n ; m ≤n ; $m \in I^+$

$$\dot{x} = f(t, x, u),$$

$$t \in R, x \in R^n, u \in R^m \qquad ....(1)$$

where $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}$$

The function f is called the vector field. It will generally be assumed that f is continuous with continuous partial derivatives with respect to t and x. f may be assume continuous differentiability with respect to u.The variables x1,…,xn are called state variables, where $x \in R^n$ the variables u1,…,um are called control variables, where $u \in R^m$ and the function $f : R \times R^n \times R^m \to R^n$ is a vector field.

- Boundary condition: the initial and final (terminal) time are denoted by t0 and tf, respectively, and the state vector is constrained to satisfy the initial condition.

$$x(t_0) = x_0 \qquad … (2)$$

where x0 is a known vector of initial conditions.

- Cost function: the cost function or performance index J(x(t),u(t)), will in general consist of two terms,

$$J(x(t),u(t)) = \Phi(x(t_f)) + \int_{t_0}^{t_f} F(t, x(t), u(t))dt \qquad ...(3)$$

where $\Phi(x(t_f))$ is the terminal cost and the integral part of the cost is a cost associated with the state and control trajectories. The functions $\Phi : R^n \to R$ and $F : R \times R^n \times R^m \to R$ are assumed to be continuously differentiable with respect to arguments.

2- B-spline Polynomials[4]

B-splines are the standard representation of smooth non-linear geometry in numerical calculation. Schoendery first introduced the B-spline in 1949. He defines the basis functions using integral convolution. B-spline means spline basis and letter

B in B-spline stands for basis. Higher degree basis functions are given by convolution of multiple basis functions of one degree lower.

In the mathematical subfield of numerical analysis a B-spline is a spline function which has minimal support with respect to a given degree, smoothness, and domain partition. A fundamental theorem states that every spline function of a given degree, smoothness and domain partition, can be represented as a linear combination of B-spline of that same degree and smoothness, and over that same partition. The term B-spline was coined by Isaac Jacob Schoenberg short for basis spline.

In the computer science subfields of computer-aided design and computer graphics the term B-spline frequently refers to a spline curve parameterized by spline functions that are expressed as linear combinations of B-spline (in the mathematical sense above).

Definition (1):[8]

Given m+1 knots t i in [0,1] with

$t_0 < t_1 < t_2 < \ldots < t_m$

a B-spline of degree n is a parametric curve $B:[0,1] \rightarrow R^2$ composed of basis B-spline of degree n

$$B(t) = \sum_{i=0}^{m+1} P_i B_{i,n}(t) \qquad t \in [0,1]$$

The Pi i=0,1,…,m+1 are called control points or anchor points or de Boor points. A polygon can be constructed by connecting the de Boor points with lines starting with P0 and finishing with Pn this polygon is called the de Boor polygon.

The m-n basis B-spline of degree n can be defined using the Cox-de Boor recursion formula

$$B_{k,0}(t) = \begin{cases} 1 & \text{if} \quad t_k \le t \le t_{k+1} \\ 0 & \text{otherwise} \end{cases}$$

$$B_{k,n} = \frac{t - t_k}{t_{k+n} - t_k} B_{k,n-1}(t) + \frac{t_{k+n+1} + t}{t_{k+n+1} - t_{k+1}} B_{k+1,n-1}(t)$$

When the knots are equidistant we say the B-spline is uniform otherwise we call it non-uniform.

The B-spline can be defined in another way like:

$$B_{k,n}(t) = \binom{n}{k} t^k (1-t)^{n-k}$$

for i=0,1,…,n

where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

There are n+1 nth degree B-spline polynomials for mathematical convenience, we usually set $B_{k,n}=0$ if $k<0$ or $k>n$.

These polynomials are quite easy to write down: the coefficients $\binom{n}{k}$ can be obtained from PASCAL's triangle; the exponent on the t term increases by one as k increases; and the exponent on the (1-t) term decreases by one as k increases.

Constant B-spline $B_{k,0}(t)$ [5]

The constant B –spline is the simplest spline. It is defined on only one knot span and is not even continuous on the knots. It is a just indicator function for the different knot spans.

$$B_{k,0}(t)=1=\begin{cases}1 & \text{if} \quad t_k \le t < t_{k+1}\\ 0 & \text{otherwise}\end{cases}$$

(2.1) Linear B-spline Bk,1(t) [5]

The linear B-spline is defined on two consecutive knot spans and is continuous on the knots, but not differentiable.

B0,1(t)=1-t

B1,1(t)=t

(2.2) Quadratic B-spline Bk,2(t)[5]

Quadratic Bspline with uniform knot–vector is a commonly used form of B-spline. The blending function can easily be recalculated, and is equal to each segment in this case

B2,2(t)=t2

B1,2(t)=2t(1-t)

B0,2(t)=(1-t)2

(2.3) Cubic B-spline Bk,3(t) [5]

Cubic B-spline with uniform knot-vector is the most commonly used form of B-spline. The blending function can easily be recalculated and is equal to each segment in this case

B3,3(t) =t3

B2,3(t) =3t2 (1-t)

B1,3(t) =3t (1-t)2

B0,3(t) =(1-t)3

(2.4) The Main Properties of B-spline

Some properties of B-spline polynomials are given throughout the following subsections.

Recursive Definition of the B-spline Polynomials

B k, n(t)=(1-t) B k ,n-1(t) +tBk-1 ,n-1(t)

The B-spline Polynomials Form a Partition of Unity

$$\sum_{k=0}^{n} B_{k,n}(t)=\sum_{k=0}^{n-1} B_{k,n-1}(t)=\sum_{k=0}^{n-2} B_{k,n-2}(t)=\ldots\sum_{k=0}^{1} B_{k,1}(t)=(1-t)+t=1$$

Express B-spline Polynomials of Degree n-1 In terms of n Linear Combination of B-spline Polynomials of Degree n.

$$B_{k,n-1}(t)=\binom{n-1}{k}\left[\frac{1}{\binom{n}{k}}B_{k,n}(t)+\frac{1}{\binom{n}{k+1}}B_{k+1,n}(t)\right]$$

$$=(\frac{n-k}{n})B_{k,n}(t)+(\frac{k+1}{n})B_{k+1,n}(t)$$

Converting from the B-spline Basis to Power Basis

$$B_{k,n}(t)=\sum_{i=k}^{n}(-1)^{i-k}\binom{n}{i}\binom{i}{k}t^{i}$$

Derivatives:

$$\frac{d}{dt}B_{k,n}(t)=n(B_{k-1,n-1}(t)-B_{k,n-1}(t))$$

for $0 \le k \le n$

A Matrix Representation for B-spline polynomials

In many applications a matrix formulation for the B-spline polynomials is useful. Those are straightforward to develop if one only looks at a linear combination of the B-spline basis functions.

B(t)=c0 B0,n (t)+ c1 B1,n (t)+…+cn Bn,n (t)

It is easy to write this as a dot product of two vectors

$$B(t)=\left[B_{0,n}(t) \quad B_{1,n}(t)\ldots B_{n,n}(t)\right]\begin{bmatrix}c_0\\ c_1\\ \vdots\\ c_n\end{bmatrix}$$

which can be converted to

$$B(t)=\begin{bmatrix}1 & t & t^2 & \ldots & t^n\end{bmatrix}\begin{bmatrix}b_{0,0} & 0 & 0 & \ldots & 0\\ b_{1,0} & b_{1,1} & & \ldots & 0\\ b_{2,0} & b_{2,1} & b_{2,2} & \ldots & 0\\ \vdots & \vdots & \vdots & \ldots & \vdots\\ b_{n,0} & b_{n,1} & b_{n,2} & \ldots & b_{n,n}\end{bmatrix}\begin{bmatrix}c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix}$$

where the bi,j are the coefficients of the power basis that are used to determine the respective B-spline polynomials.

The matrix in this case is lower triangular .In the quadratic case (n=2), the matrix representation is

$$B(t)=\begin{bmatrix}1 & t & t^2\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ -2 & 2 & 0\\ 1 & -2 & 1\end{bmatrix}\begin{bmatrix}c_0\\ c_1\\ c_2\end{bmatrix}$$

and in the cubic case (n=3) the matrix representation is

$$B(t)=\begin{bmatrix}1 & t & t^2 & t^3\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\ -3 & 3 & 0 & 0\\ 3 & -6 & 3 & 0\\ -1 & 3 & -3 & 1\end{bmatrix}\begin{bmatrix}c_0\\ c_1\\ c_2\\ c_3\end{bmatrix}$$

Initial and Final Values[7]

The initial and final values of B-spline polynomials at t=0 and t=1 respectively are given

The initial values of Bk,n(t), $0\le t\le 1$ for n=0 ,1,2,3 … are

$$B_{k,n}(0)=0 \qquad k=1,2,\ldots n$$

$$B_{0,n}(0)=1$$

$$\dot{B}_{0,n}(0)=-n$$

$$\dot{B}_{1,n}(0)=n$$

and the final values are

Bk,n(1)=0 k=0,1,...,n-1

Bn,n(1)=1

The Integration Property:[7]

For n=0,1,… and k=0,1,…,n , we have

$$\int_0^1 B_{k,n}(t)dt=\frac{1}{n+1}$$

Differentiation Property:[7]

The mth derivative of B-spline polynomials Bk,n(t) is given by

$$B_{k,n}^{m}(t)=\frac{n!}{(n-m)!}\sum_{i=0}^{m}(-1)^{i}\binom{m}{i}B_{k+i-m,n-m}(t)$$

The Product Property [7]

The product of two B-splines polynomials Bi,n(t) and Bj,m(t) of degrees n and m respectively can be written as B-spline polynomial of degree (n+m)and given by the following

For n,m=0,1,…, i=0,1,…n and j=0,1,…,m; we have

$$B_{i,n}(t)*B_{j,m}(t)=\frac{\binom{n}{i}\binom{m}{j}}{\binom{n+m}{i+j}}B_{i+j,n+m}(t)$$

3- State Vector Parameterization (SVP) for Solving Linear Quadratic Optimal Control LQOC Problem [1]

Consider the LQOC problem

$$J=\int_0^{t_f}(x^TQ\,x+u^TR\,u)dt \qquad \ldots(4)$$

subject to the linear system state equations

$$\dot{x}=Ax+Bu \qquad \ldots(5)$$

$$x(0)=x_0 \qquad \ldots(6)$$

where $x\in R^n, u\in R^m$,m ≤ n ,A and B are n × n and n × m real valued matrices respectively while Q is n× n positive semi definite matrix,

<xTQx>≥0 and R is m × m positive definite matrix uTRu >0 unless u(t)=0.

Remark (1):

1• In the following procedure, we will always assume that the matrix B is nonsingular; however when B is singular it can be approximated by a nonsingular matrix A such that $\|B-A\|_2 < e \quad , e$ is a small positive real number.

2• The Hessian matrix H. will be assumed to be positive definite matrix and thus every local minima are global minima.

3• In this work we will be concerned with the QOC problems associated with finite time of minimizing a running cost or performance index subject to linear control dynamics.

The idea of the state parameterization, using the B-spline polynomials as a basis function, is to approximate the state variables as follows:

$$x(t) \approx \sum_{i=0}^{n} a_i B_{i,n}^{*}(t) \qquad (7)$$

Where, a_i are the unknown parameters. The control variables $u(t)$ are determined from the system state equation as function of the unknown parameters of the applied parameterization technique. Two case are considered, if the numbers of states and control variables are equal to n=m, that each state variable will be approximated by finite length polynomials series and control vector is obtained as function of the state variables. If the number of the state variables is greater than the number of control variables n>m, in this case, a set of the state variables is approximated which will enable us to find the remaining state variables and control variables as functions of this set.

First, when n=1 , we have

$$x(t) \approx \sum_{i=0}^{1} a_i B_{i,1}(t) \qquad 0 \le t \le 1 \qquad (8)$$

Substitute $B_{0,1}(t), B_{1,1}(t)$ into eqn.(7) to get

$$x(t) = \sum_{i=0}^{1} a_i \binom{1}{i} t^i (1-t)^{1-ii} \qquad (9)$$

The control points a_i i=0,1 can be evaluated as follows:

Put t=0 and t=1 into (9) to get a_0, a_1 respectively, i.e

$$x(0) = a_0 \qquad , \mathrm{x}(0) = \mathrm{x}_0 \qquad (10)$$

$$x(1) = a_1 \qquad (11)$$

And differentiation (9) with respect to t and put t=0 i.e

$$\frac{dx(t)}{dt}\Big|_{t=0} = -a_0 + a_1 \qquad (12)$$

Substitute eqn.(12) into eqn.(5)

$$-a_0 + a_1 = u(0) \qquad (13)$$

Rewrite eqn.(11)and (13) in the matrix form as

$$D = \begin{bmatrix} 1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}, \qquad \mathrm{C} = \begin{bmatrix} \mathrm{a}_1 \\ \mathrm{u}(0) \\ \mathrm{x}(1) \end{bmatrix}$$

$$\text{and} \qquad \mathrm{F} = \begin{bmatrix} 0 \\ \mathrm{x}(0) \end{bmatrix} \qquad \ldots(14)$$

Or DC=F

from which we get the following system

Or DC=F

D is singular matrix, we will use **theorem** (Existence of the Moor-Penrose Inverse)[3].

To find the inverse matrix of D

From which we get the following system

$$C = D^{+}F \qquad (15)$$

Finally, Gauss elimination procedure is used to solve the above system to find

c_0, c_1, c_2

when n=2 or 3 or ….. the same step follow

4- Examples of OC Problems:[2]

In this section the performance of the proposed methods discussed in the previous section will be compared using example.

Example

Minimize

$$J = \int_0^1 \left(x^2 + u^2\right)dt \qquad \dots (16)$$

subject to

$$\dot{x} = u \qquad x(0) = 1 \quad \dots \ (17)$$

● x(t) is approximated by 3 rd order B-spline series of unknown parameters i.e.,

$$x(t) \approx \sum_{i=0}^{3} a_{i1} B_{i,3}(t) \qquad \dots (18)$$

where F=(1 0 0 0), b=(1) and a=(a01 a11 a21 a31)T

The following optimal parameters a can be found

a = (1 0.7482852 0.65006441 0.64805174)T

Substitute these optimal values into (18) to evaluate the approximate optimal value J* .Here the optimal value is found to be J*=0.76159967.

Moreover ,the optimal trajectory and the optimal control are given by:

x(t)=(1 0.7482852 0.65006441 0.64805174) B(t)

u(t)=(-0.75515243 -0.44815371 -0.19844891 -0.006038) B(t)

where B(t)= (B0,3(t) B1,3(t) B2,3(t) B3,3(t))T

Table (1) lists the values of the optimal trajectory x(t) and the optimal control u(t) obtained by the algorithm SVPB with N=3 as well as the absolute error .

This problem is also solved by expanding x(t) into different orders B-spline series with N=4,5 and 6, the optimal values of J* for each case are listed in Table (2),from which ,it can be concluded that as N increases ,we get more accurate value for J.

From table (.2) we can conclude that the sequence Jk converges to J*=0.761594156 super linearly i.e.

$$\left|J^{(k)} - J^{*}\right| \to 0$$

for sufficiently large k

Conclusions

The proposed method in this work have the following advantages: Easy method of approximation; no integration of the state equations or costate equations are needed;

References

[1] Al-Aisawi, H. S.; Computational Methods for Solving Optimal Control Problems, Ph.D.Thesis, University of technology, 2006.

[2] Al-Rawi , S. N.; On The Numerical Method For Solving Some Continuous Optimal Control Problem, Ph.D. Thesis, University of Mustansiriya, 2004.

[3] Al-Heety, F.A.; Methods For Computing Generalized Inverse Of A Matrix, Ms.C. Thesis, MSU,1982

[4] B-Spline Curves, Internet, Lesson (15),Nov.16,2001. http://designer.mech.yzu.edu.tw/

[5] B-Spline Curves, Internet. www.math.hmc.edu/gu/math.142/mellon/Application-to-GAGD/B-pline-curves.html-9k,2002

[6] Hsu, Y. L., B-Spline Curves.2002, Internet. http://designer.mech.yzu.edu.tw/

[7] Hassen S.S.;Numerical Solution and Algorithms for Oplimal Control Problems,Ms.C. Thesis, University of technology 2006

[8] Prochazkova,J.,"Derivative of B-spline Function",25.konference O Geometrii A Pocitacoue Grafice,2000

Table (1) Approximate Values of the Trajectory and Control with N=3

N	The Optimal Value Algorithm	Absolute Error |Jexact-Japp|
4	0.76159416	4×10-9
5	0.761594156	10-9
6	0.761594156	0.0000000

Table (2) B-spline polynomial of order 4,5and 6

t	3 rd order B-spline x(t)	u(t)	Absolute Error |x(t)exact-x(t)app |	|u(t)exact-u(t)app|
0	1	-0.75515243	0	0
0.1	0.92903244	-0.66477163	3.1428×10-4	6.4417×10-3
0.2	0.86693109	-0.57782847	2.0066×10-4	4.6692×10-4
0.3	0.81335216	-0.49432294	6.5477×10-5	2.2876×10-3
0.4	0.76795191	-0.41425505	2.9389×10-4	2.7195×10-3
0.5	0.73038657	-0.33762479	3.7626×10-4	1.6690×10-3
0.6	0.70031237	-0.26443216	2.8120×10-4	7.3250×10-5
0.7	0.67738555	-0.19467717	5.044×10-5	1.7576×10-3
0.8	0.66126234	-0.12835981	2.0372×10-4	2.6685×10-3
0.9	0.651599	-0.06548009	3.0175×10-4	2.1169×10-3
1	0.64805174	-0.00603710	2.6310×10-6	2.5300×10-6
J		0.76159967		| Jexact -Japp |=5.514×10-6