2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Fast Trajectory Planning via the B-spline Augmented Virtual Motion Camouflage Approach

Gareth Basset, Yunjun Xu, and Ni Li

Abstract—**Nonlinear constrained optimal trajectory control is an important and fundamental area of research that continues to advance. This paper proposes an improved version of the bio-inspired virtual motion camouflage (VMC) method, which is based on the natural phenomenon of motion camouflage. Currently, the VMC method offers fast solutions that fall close to the optimal solution, but the solution optimality is affected by a selected and fixed search space that is defined by a fixed polynomial type prey motion. To increase the flexibility of the search space and thus improve the solution optimality, this paper proposes using B-spline curves to represent the prey motion, in which the control points and thus the prey motion can be optimized. It is expected that the proposed B-spline augmented VMC method will improve the solution optimality without sacrificing the CPU time too much. A minimum time obstacle avoidance robot problem will be simulated to demonstrate the capabilities of the algorithm.**

Index Terms—**optimal control, virtual motion camouflage, B-spline**

I. Introduction

Nonlinear constrained optimal control that considers equality and inequality constraints on state and control variables is an important and fundamental area in dynamical systems' path and trajectory design. Many methodologies have been developed for a myriad of applications, and new developments continue to emerge in this active field of research. In addition to many heuristic methodologies [1]-[3] that solve for global optimal solutions, and hybrid methods [11] that combine heuristic approaches with other methods, many methods focus on finding a local optimum or improving upon it and can be generally grouped as mathematical programming

Two often used mathematical programming methods are (1) the calculus of variations (CoV) with Pontryagin's Minimum Principle (PMP) approach [4]-[6], and (2) direct collocation (DC) with nonlinear programming (NLP) approach [7],[8],[18]-[22]. Both methods have their own advantages and disadvantages. A more comprehensive literature review can be found in [27][28].

To achieve the optimal or very-close-to optimal solution rapidly, a virtual motion camouflage (VMC) method has been proposed recently [12]. Inspired by the biological motion known as motion camouflage, the VMC method dramatically reduces the dimension of the achieved NLP formulation. A dimension reduced search space is defined using a selected virtual trajectory called the prey motion (which is a virtual one and selected according to boundary conditions) and an optimized reference point. Over this subspace, the optimal or near optimal trajectory is found through optimizing the path control parameters (PCPs) for each of the optimization iterations. Because the problem is solved in a reduced search space, the computational time is significantly reduced. The dynamic equation and boundary constraints are satisfied by using the differential flatness or differential inclusion technique [15] and the necessary condition derived.

Despite these advantages, the VMC method might only be able to achieve suboptimal (but close to the optimal solution) for general cases if the dimension of the "position" state of the dynamics is larger than one. One promising approach to improve the optimality of the solution is to increase the flexibility of the prey motion and at the same time not significantly increase the problem dimension. In this paper, the B-spline technique will be applied and the VMC algorithm will be modified accordingly.

The paper is organized as follows. First, the problem definition will be listed. The regular VMC approach will then be overviewed, followed by an overview of the necessary conditions. Section III will introduce the B-spline augmented VMC algorithm. The basics of B-spline curves are discussed, and then linked to the VMC method through modified necessary conditions and the augmented algorithm itself. An analysis on the dimension and optimality of the augmented algorithm is then presented. Section IV provides a simulation example to demonstrate the capabilities of the new algorithm. Finally, conclusions are given in Section V.

II. Problem Definition and VMC Methodology

In this section, the problem definition will be provided. After this, the regular VMC approach [12],[17],[29] will be discussed briefly.

A. Problem Definition

(**P1**) For a typical nonlinear constrained optimal control problem [13], there exists a cost function

$$J=\varphi\left[\boldsymbol{x}\left(t_f\right),t_f\right]+\int_{t_0}^{t_f}L\left(\boldsymbol{x},\boldsymbol{u},t\right)dt \qquad (1)$$

that needs to be minimized (or maximized) through a set containing state $\boldsymbol{x}\in\Re^{n\times 1}$ and control $\boldsymbol{u}\in\Re^{m\times 1}$ variables and (if it is free) the final time t_f. The system is subject to inequality constraints

$$\boldsymbol{g}\left(\boldsymbol{x},\boldsymbol{u},t\right)\le 0,\quad \boldsymbol{g}\in\Re^{p\times 1} \qquad (2)$$

and equality constraints

$$\boldsymbol{h}\left(\boldsymbol{x},\boldsymbol{u},t\right)=0,\quad \boldsymbol{h}\in\Re^{q\times 1} \qquad (3)$$

The equality constraints include the boundary conditions

This work is partially funded by the National Science Foundation CMMI Program (#0939093).

Y. Xu, G. Basset, and N. Li are with the Department of Mechanical, Materials, and Aerospace Engineering, University of Central Florida, Orlando, FL 32816 USA. Phone: 407-823-1745; Fax: 407-823-0208; (email: yunjunxu@mail.ucf.edu).

978-1-61284-799-3/11/$26.00 ©2011 IEEE

$$\boldsymbol{\psi}\left[\boldsymbol{x}(t_0),\boldsymbol{x}(t_f),t_0,t_f\right]=0,\quad \boldsymbol{\psi}\in\Re^{l\times1} \tag{4}$$

and the dynamic equations of motion

$$\dot{\boldsymbol{x}}=\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},t),\quad \boldsymbol{x}\in\Re^{n\times1},\quad \boldsymbol{u}\in\Re^{m\times1} \tag{5}$$

In this paper, it is assumed that the state x can be organized into two portions: the "position" state $\boldsymbol{x}_a\in\Re^{n_a\times1}$ and the "state rate" $\boldsymbol{x}_{sr}\in\Re^{(n-n_a)\times1}$. The "state rate" can also be seen as the "remaining state." Therefore, the equations of motion in Eq. (5) can be rewritten as $\dot{\boldsymbol{x}}_a(t)=\boldsymbol{f}_a(\boldsymbol{x},t)$ and $\dot{\boldsymbol{x}}_{sr}(t)=\boldsymbol{f}_{sr}(\boldsymbol{x},\boldsymbol{u},t)$. When applying the differential inclusion technique [15], the "state rate" $\boldsymbol{x}_{sr}$ can be written as $\boldsymbol{x}_{sr}=\boldsymbol{f}_a^{-1}(\boldsymbol{x}_a,\dot{\boldsymbol{x}}_a,t)$ and the control $\boldsymbol{u}$ can be written as $\boldsymbol{u}=\boldsymbol{f}_{sr}^{-1}(\boldsymbol{x},\dot{\boldsymbol{x}}_{sr},t)$. These injective mappings can be solved either explicitly or implicitly through an iterative scheme.

B. VMC Approach

The virtual motion camouflage (VMC) method was inspired by the stealth strategy called "motion camouflage" (MC) observed in mating hoverflies [14]. In MC, a moving aggressor is attempting to chase down a moving prey without being noticed by the prey. In the VMC framework, the aggressor position $\boldsymbol{x}_a(t)$ is confined by the following variables: the prey trajectory $\boldsymbol{x}_p(t)$, the selected reference point $\boldsymbol{x}_r(t)$, and what is called the path control parameter (PCP) $\nu(t)$. The PCP is what determines where the aggressor position is placed as

$$\boldsymbol{x}_a=\boldsymbol{x}_r+\nu\left(\boldsymbol{x}_p-\boldsymbol{x}_r\right) \tag{6}$$

The reference point is considered fixed over time but can be optimized, so the derivatives of the aggressor position, i.e., the "position" state, can be derived as

$$\dot{\boldsymbol{x}}_a=\dot{\nu}\left(\boldsymbol{x}_p-\boldsymbol{x}_r\right)+\nu\dot{\boldsymbol{x}}_p \tag{7}$$

and

$$\ddot{\boldsymbol{x}}_a=\ddot{\nu}\left(\boldsymbol{x}_p-\boldsymbol{x}_r\right)+\nu\ddot{\boldsymbol{x}}_p+2\dot{\nu}\dot{\boldsymbol{x}}_p \tag{8}$$

(**P2**) Through the steps described above, the system can now be rewritten as follows. For the regular VMC method, there exists a cost function

$$J=\varphi\left[\nu,\dot{\nu},...t_f\right]+\int_{t_0}^{t_f}L\left(\nu,\dot{\nu},...,t\right)dt \tag{9}$$

that needs to be minimized (or maximized) through a set containing the PCPs, their derivatives, the reference point, and (if it is free) the final time t_f. The system is subject to inequality constraints

$$\boldsymbol{g}\left(\nu,\dot{\nu},...t\right)\le 0,\quad \boldsymbol{g}\in\Re^{p\times1} \tag{10}$$

The dynamic equation equality constraints are already taken into account by representing $\boldsymbol{x}_{sr}$ and $\boldsymbol{u}$ in terms of $\boldsymbol{x}_p$, $\boldsymbol{x}_r$, ν, and their derivatives. Boundary conditions, meanwhile, are also taken into account. Therefore, there are no explicit equality constraints.

<u>Remark 1</u>: When the system of **P1** was converted into **P2**, the search space is transferred from the full space into a search space that is defined by the prey motion.

P2 can be solved through an NLP algorithm by discretizing the PCP history $\nu(t)$ into $i=0,...,N$ nodes. Several discretization schemes [18]-[22] are available to perform the step of discretization. This paper approximates the discretized PCP history using the Legendre interpolation polynomials [8] as

$$\boldsymbol{\nu}\left(\overline{t}\right)\approx\sum_{i=0}^{N}\nu_i\phi_i\left(\overline{t}\right) \tag{11}$$

where $\overline{t}=(2t-t_f-t_0)/(t_f-t_0)\in\left[-1,1\right]$ is the zeroes of $\dot{L}_N$, the derivative of the Legendre polynomial L_N. The base functions $\phi_i(\overline{t})$, $i=0,...,N$ are the Lagrange interpolating polynomials of order N.

Furthermore, the PCP derivatives can be found in terms of the PCP vector. The equation

$$\partial^k\boldsymbol{\nu}/dt^k=\left[2/\left(t_f-t_0\right)\right]^k\boldsymbol{D}^k\boldsymbol{\nu} \tag{12}$$

finds the k^{th} order derivatives in the original time scale t. Here, $\boldsymbol{D}$ is the differentiation matrix (Fahroo and Ross 2001).

(P3) The VMC framework in **P2** can be rewritten in discretized form as follows. The cost function

$$J=\varphi\left(\nu,t_f\right)+\left(\frac{t_f-t_0}{2}\right)\sum_{k=0}^{N}L\left(\nu\right)\omega_k \tag{13}$$

needs to be minimized (or maximized) according to the PCP vector $\boldsymbol{\nu}=[\nu_k]_{k=0,1,...,N}$ and (if it is free) the final time t_f. The variables ω_k are the weights for the k^{th} node. The system is subject to inequality constraints

$$\boldsymbol{g}\left(\nu,t\right)\le 0,\quad \boldsymbol{g}\in\Re^{p\times1} \tag{14}$$

III. B-spline Augmented VMC Algorithm

A. Basics of B-spline [23]

Polynomials have proven to be very useful in representing or approximating curves. Despite their ease of use, however, their main drawback is that they can be very inflexible on large intervals, which can generate wild oscillations especially for high order curves [23]. Spline functions remedy this by taking piecewise polynomials and connecting them together while maintaining some degree of global smoothness.

For example, function $f(t)$ is represented by a B-spline curve of degree d as

$$f(t)=\sum_{i=0}^{n_{cp}}B_{i,d}\left(t\right)P_i \tag{15}$$

where $B_{i,d}(t), i=0,...,n_{cp}$ are the d^{th} degree basis functions, $P_i, i=0,...,n_{cp}$ are the control points, and $n_{cp}+1$ is the number of control points. The curve is generated over a time span $t\in\left[t_0,t_f\right]$. Defined on this same time span is what's known as the knot vector, which is

$$\tau=\left\{\underbrace{t_0,...,t_0}_{d+1},\tau_{d+1},...,\tau_{k-d-1},\underbrace{t_f,...,t_f}_{d+1}\right\} \tag{16}$$

with length $k+1$. The knots τ are the time points (or breakpoints) on time span t where the piecewise polynomials are linked together to form the B-spline curve. Naturally, the knots must be non-decreasing, i.e., $\tau_i\le\tau_{i+1}$.

There are a few types of knot vectors available; the vector shown in Eq. (16) and used in this paper is called the non-periodic knot vector. Here, the initial knot $\tau_0 = t_0$ and final knot $\tau_k = t_f$ are repeated with multiplicity $d+1$, and a B-spline curve will generally only interpolate through a control point for this type of knot vector. In this case, the curve's two endpoints will interpolate through the initial and final control points.

The B-spline basis functions are calculated recursively for each time t. First, the zero-degree functions are calculated as

$$B_{i,0}(t) = \begin{cases} 1 & if\, \tau_i \le t < \tau_{i+1} \\ 0 & otherwise \end{cases} \tag{17}$$

Then the remaining basis functions are calculated as

$$B_{i,m}(t) = \frac{t-\tau_i}{\tau_{i+d}-\tau_i} B_{i,m-1}(t) + \frac{\tau_{i+d+1}-t}{\tau_{i+d+1}-\tau_{i+1}} B_{i+1,m-1}(t), \quad m = 1,\ldots,d \tag{18}$$

up to the d^{th} degree basis functions, which are used in Eq. (15). The j^{th} derivative of the d^{th}-degree basis functions can be found recursively using

$$B_{i,d}^{(j)}(t) = d\left(\frac{B_{i,d-1}^{(j-1)}}{\tau_{i+d}-\tau_i} - \frac{B_{i+1,d-1}^{(j-1)}}{\tau_{i+d+1}-\tau_{i+1}} \right) \tag{19}$$

The number of control points is related to the degree of the spline and the number of knots as

$$n_{cp} = k - d - 1 \tag{20}$$

It is worth noting that the B-spline curve is used for each of the prey motion components, and the description in this section just represents one of them.

B. Modified Necessary Boundary Conditions

When B-spline curves are used to define the prey motion, the boundary conditions derived here can be used to solve for certain control points or control point components. Here the necessary conditions are derived for one set of boundary conditions called BC1: Fixed initial and final $\boldsymbol{x}_a$.

Lemma 1. When the initial and final "position" states are known, the first and the last control points for i^{th} component, $i = 1,\ldots n_a$, must satisfy the equations

$$P_{i,0}B_{0,d}(t_0) + P_{i,n_{cp}}B_{n_{cp},d}(t_0) = x_{a,i,0} - \sum_{k=1}^{n_{cp}-1} P_{i,k}B_{k,d}(t_0),\ i = 1,\ldots n_a \tag{21}$$

and

$$P_{i,0}B_{0,d}(t_f) + P_{i,n_{cp}}B_{n_{cp},d}(t_f) = x_{a,i,N} - \sum_{k=1}^{n_{cp}-1} P_{i,k}B_{k,d}(t_f),\ i = 1,\ldots,n_a \tag{22}$$

In this Lemma, $P_{i,k}$ is the k^{th} control point for the i^{th} direction of the prey motion.

Proof. The initial and final positions of the prey motion are selected to equal to, respectively, the initial and final aggressor positions by selecting $v_0 = v_N = 1$. This gives us the necessary conditions $\boldsymbol{x}_{p,0} = \boldsymbol{x}_{a,0}$ and $\boldsymbol{x}_{p,N} = \boldsymbol{x}_{a,N}$. Since the prey motion is represented by the B-spline curve in Eq. (15), this obtains the equations

$$\begin{aligned} \sum_{k=0}^{n_{cp}} P_{i,k}B_{k,d}(t_0) &= x_{a,i,0} \\ \sum_{k=0}^{n_{cp}} P_{i,k}B_{k,d}(t_f) &= x_{a,i,N} \end{aligned}, \quad i = 1,\ldots,n_a \tag{23}$$

In Eq. (23), the initial and final control points, $P_{i,0}$ and $P_{i,N}, i = 1,\ldots,n_a$, are calculated instead of optimized. Rearranging the equations gives us Eqs. (21-22). □

<u>*Remark 2*</u>*:* When using a non-periodic knot vector, as in Eq. (16), $B_{k,d}(t_0) = 0,\ k = 1,\ldots,n_{cp}$ and $B_{k,d}(t_f) = 0,\ k = 0,\ldots,n_{cp}-1$. Therefore, $P_{i,0} = x_{a,i,0}$ and $P_{i,n_{cp}} = x_{a,i,N}$.

<u>*Remark 3*</u>*:* For BC1, the following parameters are calculated: $v_k, k = 0, N$, and $P_{j,k}, k = 0, n_{cp},\ j \in [1,\ldots,n_a]$; and the following parameters are optimized: $\boldsymbol{x}_r$, $v_k, k = 1,..,N-1$, and $P_{j,k}, k = 1,\ldots,n_{cp}-1,\ j \in [1,\ldots,n_a]$.

(P4) Through the steps mentioned in Section II and the steps above, **P1** is converted into the following dimension reduced NLP. The cost function

$$J = \varphi(v, \boldsymbol{P}, \boldsymbol{x}_r, t_f) + \left[\frac{t_f - t_0}{2}\right] \sum_{k=0}^{N} L(v, \boldsymbol{P}, \boldsymbol{x}_r, t_f)\,\omega_k \tag{24}$$

is minimized by varying the components of the PCP vector v and the control points $\boldsymbol{P}$ that are optimized (instead of calculated), as well as the reference point $\boldsymbol{x}_r$ and (if it is free) the final time t_f. The parameters to be optimized for BC1 can be found in Remark 3. The optimization is subject to inequality constraints

$$\boldsymbol{g}(v, \boldsymbol{P}, \boldsymbol{x}_r, t) \le 0 \tag{25}$$

C. B-spline Augmented VMC Algorithm

For the B-spline augmented VMC algorithm, all variables to be optimized are grouped into Set S_g. This set contains the parameters for each respective set of boundary conditions. The following algorithm below lists the detailed steps of the B-spline augmented VMC algorithm.

Algorithm 1. B-spline Augmented VMC Algorithm

Steps in the Initialization	Step 0:	Provide initial guesses for the parameters in Set S_g depending on the boundary condition.
Steps inside the NLP Iterations	Step 1:	Calculate the remaining PCPs and control points.
	Step 2:	Evaluate the performance index using Eq. (24).
	Step 3:	Evaluate the constraints using Eq. (25).
	Step 4:	If the convergence criterion is not satisfied and the maximum number of iterations hasn't reached, generate parameters in Set S_g for the next NLP iteration, and go back to Step 1. Otherwise, the optimization is a success and terminated.

D. Dimension Analysis

The first method is a "baseline" approach. In the "baseline" approach, the nonlinear constrained trajectory design problem described in Section II is formulated as an NLP via a pseudo-spectral based collocation method such as the LGL method [8]. The states and control vectors are discretized into $0,1,...,N$ nodes, and the state and control parameters at those collocation nodes are then optimized. Therefore, the dimension of the parameters is on the order of $\mathrm{O}\left[(n+m)N\right]$.

The second method, the VMC approach [29], in which the prey motion is represented by a polynomial, optimizes the discretized PCP vector (and possibly the final time and the reference point). In this approach, the parameter dimension is on the order of $\mathrm{O}(N)$.

The third method is the B-spline augmented VMC algorithm proposed in this paper. In addition to the PCPs and possibly the reference point and final time, this augmented VMC approach needs to optimize some of the control points of the B-spline which is used to represent the prey motion. Therefore, the number of parameters to be optimized is on the order of $\mathrm{O}(N + n_a n_{cp})$, where n_a is the number of "position" states and n_{cp} is the number of control points. Normally n_{cp} is much less than the number of collocation nodes N.

IV. Simulation Example

To demonstrate the enhanced capabilities of the B-spline augmented VMC approach, a two-wheel robot minimum time trajectory planning example is presented here.

A. Problem Definition

The kinematic model of the two-wheel mobile robot [26] is given by

$$\dot{x} = \upsilon\cos\theta;\ \dot{y} = \upsilon\sin\theta;\ \dot{\theta} = w \tag{26}$$

where the two wheels' midpoint $\boldsymbol{x}_a = \left[x, y\right]^T$ and the direction of the vehicle $x_{sr} = \theta$ are regarded as the state variables. Two control variables are involved as the speed v and the angular speed w, and they are respectively constrained by $|v| \le v_{\max}$ (e.g. $v_{\max} = 0.1m/s$) and $|w| \le w_{\max}$ (e.g. $w_{\max} = 135°/s$). The mission objective is to start at a position of [1,1] with an initial direction of $\theta_0 = 45°$ and move to a position of [9,9] in the minimum possible time while avoiding all obstacles. In the discretized form, the minimum time cost function is $J = 0.5\left(t_f - t_0\right)\sum_{i=0}^{N}\omega_i$. Through differential flatness, the state rate can be computed as $\theta = \tan^{-1}\left(\dot{y}/\dot{x}\right)$, while the control variables can be computed as $v = \dot{x}/\cos\left(\theta\right)$ (or $v = \dot{y}/\sin\left(\theta\right)$ if $\cos\left(\theta\right) = 0$) and $w = \left(\ddot{y}\dot{x} - \ddot{x}\dot{y}\right)/\left(\dot{x}^2 + \dot{y}^2\right)$ (if $\left(\dot{x}^2 + \dot{y}^2\right) \ne 0$).

The obstacle used in this problem is defined as: (C1) $\left(x-5\right)^2 + \left(y-5\right)^2 = 4$. For each simulation node case, the reference point is considered as a parameter to be optimized, and an initial guess of $[130, -120]$ is used. Also, an initial guess of a straight line connecting the endpoints is used for the control points.

B. Simulation Results and Discussion

Table 1 shows the results using the baseline approach, the regular VMC approach, and the B-spline augmented VMC approach. The "NURBS" used in the B-spline augmented VMC approach has a degree of three and four control points.

Several observations are apparent in these results. First, all of the VMC methods generate results that fall within 1.8% of the baseline's results. The difference percentage between the baseline's and the VMC methods' results gets smaller as the number of nodes increases. Second, compared to the baseline approach, all of the VMC methods have significantly smaller CPU runtimes. The baseline method's runtime increases noticeably as the number of nodes increases (from 3.65 to 24.06 seconds), while the VMC methods have a much smaller increase as the number of nodes increases. The regular VMC method ranges from 1.07 to 1.57, and the augmented method point ranges from 2.30 to 5.72. Third, there are noticeable differences in the results between the regular VMC and B-spline augmented VMC methods. The augmented VMC method obtains results that are much closer to the baseline solutions, while the regular VMC method achieves its results with faster runtimes. This is because the augmented VMC method optimizes more variables compared to the regular VMC method

Table 1 Simulation results of the robot minimum time collision avoidance problem (***1 obstacle***)

Algorithm	Performance	10-node	15-node	20-node	25-node
"baseline" approach	Index (s)	120.8123	120.2759	120.3392	120.3258
	CPU Time (s)	**3.6499**	**14.2157**	**31.7255**	**24.0594**
		VMC methods			
VMC	Index (s)	122.4358	121.9288	121.1358	121.0189
	Difference %	1.7536	1.3322	0.6732	0.5760
	CPU Time (s)	**1.0727**	**1.1632**	**1.3078**	**1.5735**
Augmented VMC w/ optimized ref. pt.	Index (s)	120.7515	121.5823	120.8751	120.7515
	Difference %	0.7870	1.0860	0.4450	0.3540
	CPU Time (s)	**2.2962**	**2.3382**	**4.3058**	**5.7203**

Figure 1 displays the results for the 1-obstacle 25-node. Here, a straight line prey motion is used for the regular VMC while the B-spline prey motion is used for the augmented VMC. Both VMC methods follow the path of the baseline approach very well but the augmented VMC is closer to the baseline one than the regular VMC approach. The figure also illustrates how much the prey motion changes in the augmented VMC approach to improve the performance index, as compared with the fixed straight line used in the regular VMC approach.

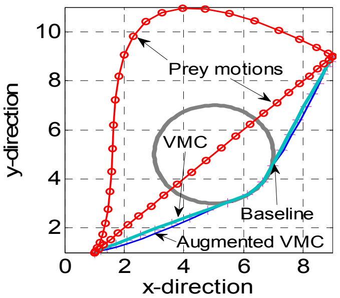

Fig. 1 One obstacle optimal trajectory

V. Conclusion

In this paper, the virtual motion camouflage method has been significantly enhanced by including a B-spline as the prey motion in solving nonlinear constrained trajectory optimal control type problems. The main advantage of the VMC method has been maintained, i.e., the problem dimension is dramatically reduced, which in practice will reduce the computational time significantly compared with baseline method. The solution searching space constructed by the prey motion is more flexible and the solution's optimality is improved significantly. The necessary conditions, as well as the dimension and optimality analyses are provided. The simulation results demonstrate the effectiveness and advantages of the new algorithm.

References

[1] Nusyirwan, I. F., and Bil, C, "Effect of uncertantines on UCAV optimisation using evolutionary programming," *IEEE Information, Decision, and Control Conference*, Adelaide, Australia, 2007.

[2] Kaneshige, J., and Krishnakumar, K., "Arificial immune system approach for air combat maneuvering," *Intelligent Computing: Theory and Applications*, Edited by Priddy, K. L. & Ertin, E, 2007, 6560, 656009-1 – 656009-12.

[3] Merchan-Cruz, E. A., and Morris, A. S, "Fuzzy-GA-based trajectory planner for robot manipulators sharing a common workspace," *IEEE Transactions on Robotics*, Vol. 22, No. 4, 2006, pp. 613-624.

[4] Lawden, D. F., "Rocket trajectory optimization: 1950 – 1963," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 4, 1991.

[5] Ocampo, C., "Finite burn maneuver modeling for a generalized spacecraft trajectory design and optimization system," *Annuals of the New York Academy of Sciences*, Vol. 1017, 2004, pp. 210-233.

[6] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., *The Mathematical theory of optimal processes*, Wiley-Interscience, New York, NY, 1962.

[7] Betts, J., "Practical methods for optimal control using nonlinear programming," *Society for Industrial and Applied Mathematics*, Philadelphia, 2001.

[8] Fahroo, F., and Ross, I. M, "Costate estimation by a legendre pseudospectral method," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 270-275.

[9] Jacobson, D. H., and Lele, M. M, "A transformation technique for optimal control problems with a state variable inequality constraint," *IEEE Transaction on Automatic Control*, Vol. 14, No. 5, 1969.

[10] Mehra, R. K., and Davis, R. E, "A generalized gradient method for optimal control problems with inequality constraints and singular arcs," *IEEE Transactions on Automatic Control*, Vol. 17, No. 1, 1972, pp. 69-79.

[11] Hristu-Varsakelis, D., and Shao, C., "Biologically-inspired optimal control: learning from social insects," *International Journal of Control*, Vol. 77, No. 18, 2004, pp. 1549-1566.

[12] Xu, Y.., "Motion camouflage and constrained suboptimal trajectory control," *2007 AIAA Guidance, Control, and Dynamics Conference*, August 20-23, 2007, Hilton Head, South Carolina.

[13] Hartl, R. F., Sethi, S. P., and Vickson, R. G., "A survey of the maximum principles for optimal control problems with state constraints," *SIAM Review*, Vol. 37, No. 2, 1995, pp. 181-218.

[14] Srinivasan, M. V., and Davey, M., "Strategies for active camouflage motion," *Proceedings of the Royal Society of London Biological Sciences*, Vol. 259, No. 1354, Jan. 23, 1995, pp. 19-25.

[15] Kumar, R. R. and Seywald H., "Should controls be eliminated while solving optimal control problems via direct methods?" *Journal of Guidance, Control and Dynamics*, Vol. 19, No. 2, 1996, pp. 418-423.

[16] Ferrante, A., and Ntogramatzidis, L., "A unified approach to the finite-horizon LQ regulator – part I: the continuous time," *45th IEEE Conference on Decision and Control*, San Diego, CA, USA, December 13-15, 2006, pp. 5651-5656.

[17] Xu, Y., and Basset, G., "Real-Time Optimal Coherent Phantom Track Generation via the Virtual Motion Camouflage Approach," 2010 AIAA Guidance, Navigation, and Control Conference.

[18] Hager, W., "Runge-Kutta methods in optimal control and the transformed adjoint system," *Numerische Mathematik*, Vol. 87, No. 2, 2000, pp. 247-282.

[19] Milam, M., Mushambi, K, and Murray, R., "A computational approach to real-time trajectory generation for constrained mechanical systems," *Conference on Decision and Control*, 2000, Sydney NSW Australia.

[20] Dai, R., "B-splines based optimal control solution," *2010 AIAA Guidance, Navigation, and Control Conference*, August 2-5, 2010, Toronto, Ontario Canada, Paper # 2010-7888.

[21] Jackiewicz, Z., and Welfert, B. D., "Stability of gauss-radau pseudospectral approximations of the one-dimensional wave equation," *Journal of Scientific Computing,* Vol. 18, No. 2, 2003.

[22] Kameswaran, S., and Biegler, L. T., "Convergence rates for direct transcription of optimal control problems using collocation at Radau points," *Computational Optimization and Applications*, Vol. 41, No. 1, 2008, pp. 81-126.

[23] Piegl, L, and Tiller, W., "The NURBS Book: Second Edition," Germany, 1997.

[24] Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1435-1440.

[25] Rao, C. V., Wright, S. J., and Rawlings, J. B., "Application of interior-point methods to model predictive control," *Journal of Optimization Theory and Applications*, Vol. 99, No. 3, 1998.

[26] Laumond, J. P., Sekhavat, S., and Lamiraux, F., "Guidelines in nonholonomic motion planning for mobile robots," *Lecture Notes in Control and Information Sciences*, LNCIS 229, New York: Springer-Verlag, 1998, pp. 1-44.

[27] Betts, J., "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998.

[28] Goerzen, C., Kong, Z, and Mettler, B., "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent and Robotic Systems*, Vol. 57, 2010.

[29] Xu, Y., "Analytical Solutions to Formation Flying System Trajectory Guidance via the Virtual Motion Camouflage Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, Sept.-Oct. 2010.