




# Persistent Monitoring by Multiple Unmanned Aerial Vehicles Using Bernstein Polynomials

Calvin Kielas-Jensen<sup>1</sup>  · Venanzio Cichella<sup>1</sup> · David Casbeer<sup>2</sup> · Satyanarayana Gupta Manyam<sup>3</sup> · Isaac Weintraub<sup>4</sup>

Received: 31 March 2020 / Accepted: 3 August 2021 / Published online: 24 August 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

A framework for monitoring a target modeled as Dubins car using multiple UAVs is proposed. The UAVs are subject to minimum and maximum speed, maximum angular rate constraints, as well as inter-vehicle safety requirements and no-fly-zones. The problem is formulated as a continuous time nonlinear optimal control problem. This problem is first simplified by using a sequential approach, which significantly reduces its complexity. Then, by defining the desired trajectories to be tracked by the UAVs as Bernstein polynomials, it is transcribed into a nonlinear optimization problem. It is shown through numerical simulations that the present approach is computationally efficient, and thus it is well suited for trajectory planning/re-planning to monitor a target of unknown speed, heading direction and unexpected detours. Moreover, the

---

Communicated by Mauro Pontani.

---

✉ Calvin Kielas-Jensen  
calvin-kielas-jensen@uiowa.edu

Venanzio Cichella  
venanzio-cichella@uiowa.edu

David Casbeer  
david.casbeer@us.af.mil

Satyanarayana Gupta Manyam  
msgupta@gmail.com

Isaac Weintraub  
isaac.weintraub.1@us.af.mil

<sup>1</sup> University of Iowa, Iowa City, IA, USA

<sup>2</sup> Control Science Center of Excellence, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433, USA

<sup>3</sup> Infoscitex Corp., A DCS company, Dayton, OH 45431, USA

<sup>4</sup> Aerospace Systems Directorate, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433, USA

proposed method guarantees satisfaction of feasibility and safety constraints for the whole planning time period, rather than only at discrete time points.

**Keywords** Multi-agent system · Target monitoring · Bernstein polynomials

## 1 Introduction

Monitoring a moving target in dynamic environments is a challenging problem with many practical applications such as studying and following marine animals [27], search and rescue [38], convoy protection [13], or tracking the spread of wildfires [5]. A collection of multiple agents can be demonstrably more powerful than a single unit of the collective [15], however, this introduces additional complexities such as scheduling and avoiding collisions. This paper addresses this problem by proposing a framework for generating optimal trajectories for multiple autonomous agents to monitor a target.

The problem of tracking a target by autonomous systems has received a great deal of attention in the research community, see [29,30] and citations therein. Methods for tracking a target with a known position include cell decomposition methods [4], path planning-based approach [24], Lyapunov-based nonlinear control [25], oscillatory control [21], differential games [14], and dynamic programming-based methods [22]. Tracking a target with unknown motion and dynamics properties requires an approach that takes the uncertainty into account. In [1] the target's motion is assumed to be described by a 2D stochastic process which is represented as 2D Brownian motion and the Markov chain approximation method is employed for numerically determining the optimal control policy. The work presented in [28] seeks to continuously keep a stochastic target in the view of a gimbaled camera while avoiding blind spots, like those produced by landing. A cutting-plane approach is presented in [34], which enables multiple vehicles to seek and detect one or more uncertain targets. In [6] a temporal and spatial discretization method is proposed to formulate the optimal target tracking problem as a nonlinear programming problem subject to turn-rate constraints of the monitoring agents. In [26] a monitoring agent is equipped with a camera on a gimbal allowing the agent to monitor the target from several different line-of-sight angles.

When dealing with multi-vehicle tracking missions in a real-world environment, control problems that arise from this application are very complex since they must take into account vehicle's dynamic constraints, inter-vehicle safety, multiple mission objectives and environmental uncertainties. These lead to optimal control problems with cost function and constraints that can be highly nonlinear, non-smooth, and non-convex, and closed-loop solutions cannot be sought. Thus, solutions to these problems must be found using numerical methods such as direct methods [31]. Direct methods are based on approximating optimal control problems as nonlinear programming problems (NLPs) using some discretization scheme [2,3,12,31]. Then, these NLPs can be solved using off-the-shelf optimization solvers (e.g., MATLAB, SNOPT, etc.). A wide range of direct methods that use different discretization schemes have been developed, which include Euler [16], Runge–Kutta [35,36], Pseudospectral [33], as well as methods based on Bernstein approximants [9,10].

In the context of the existing literature on multi-vehicle target tracking, this work differs from other approaches in a fundamental way. Rather than dealing with simplified problems and designing closed-loop (reactive) controllers, we formulate the multi-vehicle target tracking problem as a general nonlinear optimal planning problem and find a solution using a direct method. This approach allows us to consider the cooperative target tracking problem in its full complexity, and rigorously ensure satisfaction of dynamics and safety constraints and mission objectives. Furthermore, the method enhances human operators' trust through transparent and predictable operations, since the solutions are generated open loop, in one shot, for long temporal windows.

We employ the direct method based on Bernstein approximants that was initially proposed in [8]. Bernstein polynomials (BPs) were originally introduced by Sergei Natanovich Bernstein (1880–1968) as a means to develop a constructive proof of the Weierstrass theorem. Due to their slow convergence as function approximants, Bernstein polynomials were not widely used until two automotive engineers from competing companies used them as a parametric representation for automotive bodies. Today, Bernstein polynomials (sometimes referred to as Bézier curves) are frequently used in graphics software and font representations. Bernstein polynomials enjoy the benefit of low computational requirements since many of their properties can be discerned from their coefficients. For example, a Bernstein polynomial is guaranteed to reside in the convex hull defined by its coefficients, leading to safe estimates of its bounds at a low computational cost, allowing generation of trajectories in near real-time fashion. Here we exploit the properties and algorithms of Bernstein polynomials to address the problem of target tracking. This approach allows us to consider the complex scenario where the vehicles must coordinate in order to ensure continuous monitoring of a partially unknown moving target by gimbeled cameras mounted on-board the UAVs. To ensure robustness to a target's motion uncertainties, we propose an event-triggered approach to re-plan trajectories using up-to-date measurements. For more in depth discussion on the benefits of Bernstein polynomial-based approaches as compared to other direct methods for trajectory generation, we refer to [8,11,17,19].

Our proposed method comprises three main parts: sequential planning, approximation via BPs, and target prediction re-planning. Sequential planning allows each agent to plan their own trajectory which offers a lower computational expense and does not require a centralized processor. BPs allow the optimal target tracking problem, an infinite dimensional optimal control problem, to be formulated as a finite dimensional nonlinear optimization problem. The latter can be solved using off-the-shelf solvers, e.g., *fmincon*, *SciPy*, *SNOPT*, and *IPOPT*. The target prediction re-planning allows trajectories to be re-planned when the expected future position of the target varies further than a predefined distance away from the original prediction used to generate the initial optimal trajectory. A preliminary version of this problem was approached in [37].

The paper begins with the problem formulation of the trajectory generation for target tracking in Sect. 2. Section 3 presents the polynomial and sequential approximations to the problem at hand and describes both the sequential approach and the Bernstein approximation method adopted. Section 4 presents numerical results of simulated missions and Sect. 5 draws conclusions.

## 2 Trajectory Generation for Target Tracking

In what follows, vectors are denoted by bold letters, e.g.  $\mathbf{p} = [p_x, p_y]^\top$ .  $\|\cdot\|$  denotes the Euclidean norm (or magnitude), e.g.,  $\|\mathbf{p}\| = \sqrt{p_x^2 + p_y^2}$ , and  $\angle \cdot$  denotes the direction angle with respect to the  $x$  axis, e.g.,  $\angle \mathbf{p} = \tan^{-1} \frac{p_y}{p_x}$ .

The objective of this work is to maintain constant camera coverage of a ground target using multiple unmanned aerial vehicles (UAVs) traveling at a constant altitude. The UAVs are assumed to be of the fixed wing variety which allows for efficient power usage. This means that the UAVs are both nonholonomic and lack the ability to hover. The camera is assumed to be fixed in order to reduce both the cost and mechanical complexity of the UAVs; therefore, the target will only lie in the camera footprint when the UAV is facing towards the target.

Consider an example monitoring mission involving 2 UAVs which takes place over the time interval  $[0, T]$ ,  $T \gg 0$ , and let the target's position be denoted as  $\mathbf{p}_\tau$  (depicted as a red star in Fig. 1). For the safety of the monitoring vehicles, a no-fly-zone with radius  $R_T$  is imposed around the target (depicted as a red circle in Fig. 1). Similarly, an outer monitoring radius  $R_O$  represents the maximum distance at which the monitoring vehicles begin their monitoring trajectories (depicted as a black circle in Fig. 1). The value for  $R_O$  is user defined and can depend on several considerations, including the cameras being used, the number of vehicles involved in the mission, or the target's dynamics. The initial monitoring points  $I_i$  lie on the outer monitoring circle and are randomly chosen from a uniform distribution to mitigate the target predicting future positions of the UAVs. At time  $t_1$ , UAV<sub>1</sub> must reach the point  $I_1$  while also facing directly towards the target. Between times  $t_1$  and  $t_2$ , UAV<sub>1</sub> will follow the monitoring trajectory depicted as a blue dotted line between points  $I_1$  and  $F_1$ . At time  $t_2$ , UAV<sub>1</sub> will reach final point  $F_1$  and simultaneously UAV<sub>2</sub> will reach its initial point  $I_2$ . Between times  $t_2$  and  $t_3$ , UAV<sub>1</sub> will follow a trajectory (depicted as a green dotted line in Fig. 1) to its new initial point  $I_3$  while UAV<sub>2</sub> will follow its monitoring trajectory between points  $I_2$  and  $F_2$ . At time  $t_3$ , UAV<sub>1</sub> will reach its initial monitoring point  $I_3$  and UAV<sub>2</sub> will reach the end of its monitoring trajectory at point  $F_2$ . From there, the process begins anew and repeats until some final time  $T$ . The monitoring vehicles are given a fixed time interval  $T_\Delta$  to complete their trajectories, e.g.,  $t_2 = t_1 + T_\Delta$ . The fixed time interval  $T_\Delta$  is a constant value which depends on the length of the monitoring trajectory and the monitoring speed of the vehicle, i.e.,  $T_\Delta = \frac{\text{monitoring trajectory length}}{\text{monitoring speed}}$ .

This monitoring scenario can be extended to an arbitrary number of UAVs, where UAV <sub>$i$</sub>  begins its monitoring trajectory at time  $t_i = t_{i-1} + T_\Delta$ , when the previous agent, namely UAV <sub>$i-1$</sub> , completes its monitoring maneuver. To minimize agents predictability, the points  $I_i$  are selected randomly on the outer circle. Furthermore, since the agents are assumed to be equipped with non-gimballed sensors, the heading angle of the agents at  $I_i$  must be equal to the line-of-sight angle to the target. Similarly, in order to guarantee monitoring, the difference between the UAVs' heading angle and the line-of-sight angle to the target must be minimized throughout the execution of the monitoring maneuvers. The problem of trajectory generation for multi-vehicle

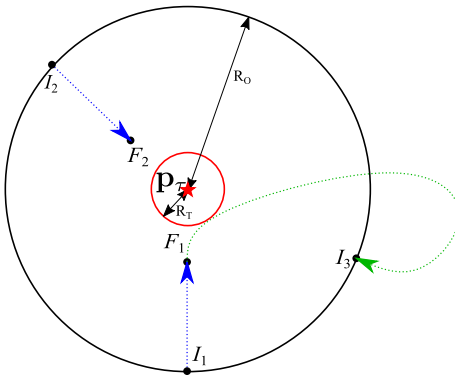


Fig. 1 Graphical depiction of the problem

continuous target monitoring missions can be formulated as a general optimal control problem, i.e.,

*Consider a mission involving  $m$  UAVs. Find  $m$  trajectories to be tracked by the UAVs over the time interval  $[t_0, t_f]$ , that minimize a cost function,  $J(\cdot)$ , subject to boundary conditions, feasible flight constraints, and mission specific constraints.*

Notice that the proposed formulation accounts for re-planning during mission execution in order to react to unpredicted events (i.e. external obstacles, target’s unknown dynamics, etc.). In fact, the above problem is defined over the time interval  $[t_0, t_f]$ , where the initial time  $t_0$  is the time at which planning occurs (different from the initial time of the overall monitoring mission,  $t = 0$ ).

In order to formulate the trajectory generation problem mathematically, let the trajectory assigned to the  $i$ th vehicle be defined as  $\mathbf{p}^{[i]} : \mathbb{R}^+ \rightarrow \mathbb{R}^2$ . Moreover, let the position of the target be defined as  $\mathbf{p}_T : \mathbb{R}^+ \times \Theta \rightarrow \mathbb{R}^2$ . The target’s position is supposed to be conditionally deterministic, i.e. there exists a stochastic variable  $\theta = [\theta_1, \theta_2]^T \in \Theta \subset \mathbb{R}^2$  such that the target’s position conditioned on  $\theta$  at time  $t$  is given by  $\mathbf{p}_T(t, \theta)$ , with dynamics governed by  $\dot{\mathbf{p}}_T(t, \theta) = [\theta_1 \cos(\theta_2), \theta_1 \sin(\theta_2)]^T$ . The variables  $\theta_1$  and  $\theta_2$  represent the stochastic speed and heading angle of the target, respectively.

We assume that the probability density of  $\theta$  over  $\Theta$  is known and is denoted by  $p_\theta : \Theta \rightarrow \mathbb{R}$ . Then, assuming that the position of the target at  $t_0$  is known, the expected position of the target at  $t > t_0$  is given by

$$\mathbb{E}[\mathbf{p}_T(t, \theta)] = \int_{\Theta} \left( \int_{t_0}^t \dot{\mathbf{p}}_T(\tau, \theta) d\tau \right) p_\theta(\theta) d\theta. \tag{1}$$

For notation simplicity, let us define  $E_{\mathbf{p}_T}(t_0, t) \equiv \mathbb{E}[\mathbf{p}_T(t, \theta)]$ , in order to highlight the dependency of the expected position of the target on the initial position (measured at  $t_0$ ). Moreover, we denote the expected line-of-sight angle to the target at time  $t$  by  $E_{\lambda^{[i]}}(t_0, t) \equiv \angle(E_{\mathbf{p}_T}(t_0, t) - \mathbf{p}^{[i]}(t))$ . The target’s position (and, therefore, the line-of-sight angle) can be measured on-line, and the planning algorithm is triggered if

the actual position of the target deviates from its expected value, computed previously when planning occurred.

Let

$$\mathcal{I}^{[i]} = \{t_1^{[i]}, t_2^{[i]}, \dots, t_{n_m}^{[i]}\}, n_m \geq 1, \quad t_0 \leq t_1^{[i]} < t_{n_m}^{[i]} + T_\Delta \leq T \quad (2)$$

be the set of time points at which UAV<sub>*i*</sub> is required to start a monitoring maneuver, where  $T_\Delta > 0$  is the pre-specified monitoring time. In order to maximize the probability that, at time  $t_j^{[i]}$ , UAV<sub>*i*</sub> is positioned along a circle of radius  $R_o$  centered around the target, the **cost function**

$$J_{I_j}(\mathbf{p}^{[i]}(t)) = \left\| \mathbf{p}^{[i]}(t_j^{[i]}) - E_{\mathbf{p}_T}(t_0, t_j^{[i]}) + \mathbf{R}_o \right\|^2 \quad (3)$$

must be minimized, where  $\mathbf{R}_o = R_o[\cos(\alpha_r), \sin(\alpha_r)]^\top$ , and  $\alpha_r \in [0, 2\pi]$  is a uniformly random angle generated before the flight trajectory is planned, in order to minimize agents’ predictability. Finally, one additional objective is to maximize the probability that, for the interval of time  $t \in [t_j^{[i]}, t_j^{[i]} + T_\Delta]$ , the heading angle of UAV<sub>*i*</sub>, denoted by  $\psi^{[i]}(t)$ , is equal to the line-of-sight angle to the target. This is captured by the following cost:

$$J_{F_j}(\mathbf{p}^{[i]}(t)) = \int_{t_j^{[i]}}^{t_j^{[i]} + T_\Delta} \left\| E_{\lambda^{[i]}}(t_0, \tau) - \psi^{[i]}(\tau) \right\|^2 d\tau. \quad (4)$$

With this setup, the cost function to be minimized is

$$J(\mathbf{p}_1(t), \dots, \mathbf{p}_m(t)) = w_1 J_{F_j}(\mathbf{p}^{[i]}(t)) + w_2 \sum_{i=1}^m \sum_{j=1}^{n_m} J_{I_j}(\mathbf{p}^{[i]}(t)), \quad (5)$$

where  $w_1, w_2 > 0$  are weights.

**Boundary conditions** on the trajectories can be imposed, such as initial position, speed, heading angle, and angular rate constraints:

$$\mathbf{p}^{[i]}(t_0) = \mathbf{p}_0^{[i]}, \quad \|\mathbf{p}^{[i]}(t_0)\| = v_0^{[i]}, \quad \psi^{[i]}(t_0) = \psi_0^{[i]}, \quad \dot{\psi}^{[i]}(t_0) = \omega_0^{[i]} \quad (6)$$

$\forall i = 1, \dots, m$ , where  $\mathbf{p}_0^{[i]}$ ,  $v_0^{[i]}$ ,  $\psi_0^{[i]}$ , and  $\omega_0^{[i]}$  are the actual position, speed, heading angle, and angular rate of UAV<sub>*i*</sub> at time  $t_0$ .

Feasible and flyable trajectories are trajectories that can be tracked by the UAVs without having them exceed pre-specified bounds on the vehicles’ states and control inputs and that ensure collision avoidance. The **feasibility and flyability constraints** considered are minimum and maximum speed, maximum angular rate, and inter-

vehicle safety:

$$v_{\min} \leq \|\dot{\mathbf{p}}^{[i]}(t)\| \leq v_{\max}, \quad (7)$$

$$\|\dot{\boldsymbol{\psi}}^{[i]}(t)\| \leq \omega_{\max}, \quad (8)$$

$$\|\mathbf{p}^{[j]}(t) - \mathbf{p}^{[i]}(t)\| \geq D_s, \quad (9)$$

$\forall t \in [t_0, t_f]$  and  $\forall i, j = 1, \dots, m, i \neq j$ , where  $v_{\max} > v_{\min} > 0$ ,  $\omega_{\max} > 0$ , and  $D_s > 0$ .

Finally, in order to prevent detection by the target, we impose **no-fly-zone constraints**, namely the trajectory cannot enter a circle of radius  $R_T$  around the target, which can be expressed as follows:

$$\|\mathbf{p}^{[i]}(t) - E_{\mathbf{p}_T}(t_0, t)\| \geq R_T, \quad \forall t \in [t_0, t_f], \quad \forall i = 1, \dots, m. \quad (10)$$

With the equations defined above, we now formulate the trajectory generation problem.

**Problem 2.1** (*Trajectory generation*) Given the prespecified parameters  $R_0$ ,  $T$ , and  $T_\Delta$  find  $m$  trajectories for the UAVs to track over the time interval  $[t_0, t_f]$  that minimize the cost function  $J(\mathbf{p}_1(t), \dots, \mathbf{p}_m(t))$  (Eq. 5) subject to boundary conditions (Eq. 6), feasible flight constraints (Eqs. 7, 8, and 9), and mission specific constraints (Eq. 10).

**Remark 2.1** Problem 2.1 is concerned with the generation of desired trajectories to be tracked by the UAVs. We consider UAVs that are equipped with controllers that stabilize the vehicles' dynamics and provide trajectory tracking capabilities (for example, see [7, 18, 32]). Notice that these capabilities are provided by most commercially available autonomous vehicle platforms. For simplicity, in this paper we assume that the on-board controllers provide ideal tracking performance, i.e. the UAVs' position and attitude coincide with the desired ones for all times. Nevertheless, the present method can be easily extended to the more realistic case on non-ideal tracking performance by properly including additional constraints in Problem 2.1.

### 3 Polynomial and Sequential Problem Approximation

This section is divided into two parts. First, we present our approach to separate the overall trajectory generation problem into different sub-problems that can be solved sequentially in a decentralized fashion, in order to handle scalability issues and the problem's uncertainties. Second, we introduce a method based on Bernstein polynomials that approximates the cost function and constraints at hand, in order to transcribe the trajectory generation problems into nonlinear programming (NLP) problems.

#### 3.1 Sequential Planning

The trajectory generation problem formulated in Sect. 2 minimizes the cost function for all vehicles simultaneously over the mission duration  $[0, T]$ . Consequently, scalability

issues arise when the time interval and the number of vehicles involved in the mission become large. To address this issue, we propose a sequential approach, wherein the trajectories for each vehicle are computed independently from the other vehicles’ trajectories. Moreover, each UAV trajectory is divided into two components, each of which is computed separately. These two components are the trajectory from the current UAV position to the outer-circle centered around the target and the *monitoring* trajectory.

Recall that Eq. (2) is the set of time instants at which UAV<sub>*i*</sub> is required to start a monitoring maneuver. Then,  $\mathcal{F}^{[i]} = \{t_j^{[i]} + T_\Delta\}_{j=1}^{n_m}$  is the set of time points at which the same UAV is required to end its monitoring maneuver. Given a monitoring mission involving *m* UAVs, each UAV is tasked with solving *n<sub>m</sub>* *flight* trajectory generation problems, and *n<sub>m</sub>* *monitoring* trajectory generation problems. The *j*th *flight* trajectory generation problem for UAV<sub>*i*</sub> is stated as follows:

**Problem 3.1** (*Flight trajectory generation*) At given time  $t_0 < t_j^{[i]}$ , find  $\mathbf{p}^{[i]}(t)$ ,  $t \in [t_0, t_f]$ ,  $t_f = t_j^{[i]}$  that minimizes the following:

$$\left\| \mathbf{p}^{[i]}(t_f) - E_{\mathbf{p}_T}(t_0, t_f) + \mathbf{R}_{i,p} \right\|^2 + \left\| E_{\lambda^{[i]}}(t_0, t_f) - \psi^{[i]}(t_f) \right\|^2, \tag{11}$$

subject to the boundary conditions described in Eqs. (6)–(10).

The *j*th *monitoring* trajectory generation problem for UAV<sub>*i*</sub> is formulated as follows:

**Problem 3.2** (*Monitoring trajectory generation*) At given time  $t_0 \geq t_j^{[i]}$ , find  $\mathbf{p}^{[i]}(t)$ ,  $t \in [t_0, t_f]$ ,  $t_f = t_j^{[i]} + T_\Delta$  that minimizes

$$\int_{t_0}^{t_f} \left\| E_{\lambda^{[i]}}(t_0, \tau) - \psi^{[i]}(\tau) \right\|^2 d\tau \tag{12}$$

subject to Eqs. (6)–(10).

Under nominal conditions, the initial time, *t*<sub>0</sub>, of the *monitoring* trajectory is equal to the final time, *t*<sub>*f*</sub>, of the previous *flight* trajectory. Similarly, the initial time of a *flight* trajectory corresponds to the final time of the previous *monitoring* trajectory. However, in the presence of *unpredicted* events, such as unexpected target detours, the (flight or monitoring) trajectory planner can be triggered, with the initial time set as the current time. In this work, re-planning is triggered when the expected value of the future position of the target deviates more than a certain distance from the original expected value. In particular, recall that  $E_{\mathbf{p}_T}(t_0, t_f)$  is the expected position of the target computed when planning occurred, while the expected position computed at current time *t* is  $E_{\mathbf{p}_T}(t, t_f)$ . If at time *t* the distance between the two values is greater than a pre-defined constant  $\epsilon$ , then re-planning is triggered.



### 3.2 Bernstein Approximation

In what follows, we present a method based on the use of BPs to transcribe Problems 3.1 and 3.2 into discrete, finite-dimensional optimization problems. For additional details on the transcription method, the reader is referred to [9]. Let the position of the  $i$ th UAV involved in the monitoring mission be defined over an arbitrary interval  $[t_0, t_f]$  by a 2-dimensional  $n$ th order BP, i.e.,

$$\mathbf{p}^{[i]}(t) \triangleq \mathbf{p}_n^{[i]}(t) = \sum_{k=0}^n \bar{\mathbf{p}}_{k,n}^{[i]} b_{k,n}(t), \tag{13}$$

where  $\bar{\mathbf{p}}_{k,n}^{[i]} \in \mathbb{R}^2$  is  $k$ th Bernstein coefficients, and  $b_{k,n}(t)$  is the Bernstein basis (see Eq. (24) in the Appendix). The derivative of  $\mathbf{p}^{[i]}(t)$ , i.e.  $\dot{\mathbf{p}}^{[i]}(t)$ , is a 2-dimensional  $n$ th order Bernstein polynomial computed as

$$\dot{\mathbf{p}}^{[i]}(t) = \sum_{\ell=0}^n \left( \sum_{k=0}^n \bar{\mathbf{p}}_{k,n}^{[i]} \mathbf{D}_{k,\ell} \right) b_{\ell,n}(t) = \sum_{\ell=0}^n \bar{\mathbf{p}}_{\ell,n}^{[i]'} b_{\ell,n}(t), \tag{14}$$

where  $\mathbf{D}_{k,\ell}$  is the  $(k, \ell)$ th entry of the  $(n + 1)$  by  $(n + 1)$  differentiation matrix  $\mathbf{D}_n = \mathbf{D}_{n-1} \mathbf{E}_{n-1}^n$  that is obtained by combining the differentiation and degree elevation properties of BPs (see Properties A.3 and A.4). Note that higher order derivatives of BPs can be computed similarly. The speed,  $v^{[i]}$ , and the heading angle,  $\psi^{[i]}(t)$ , are

$$v^{[i]}(t) = \|\dot{\mathbf{p}}^{[i]}(t)\|, \quad \psi^{[i]}(t) = \angle \dot{\mathbf{p}}^{[i]}(t), \tag{15}$$

and the angular rate,  $\omega^{[i]}(t)$ , is given by

$$\omega^{[i]}(t) = \frac{\dot{\mathbf{p}}^{[i]}(t) \times \ddot{\mathbf{p}}^{[i]}(t)}{\|\dot{\mathbf{p}}^{[i]}(t)\|^2} = \sum_{j=0}^n \bar{\omega}_{j,n}^{[i]} b_{j,n}. \tag{16}$$

The Bernstein coefficients  $\bar{\omega}_{j,n}^{[i]}$ ,  $j = 0, \dots, n$ , can be easily computed from the Bernstein coefficients of  $\dot{\mathbf{p}}^{[i]}(t)$  and  $\ddot{\mathbf{p}}^{[i]}(t)$  by using the arithmetic properties of BPs (Property A.5).

Using BPs, the cost function in Eq. (11) can be approximated as follows:

$$\left\| \bar{\mathbf{p}}_{n,n}^{[i]} - E_{\mathbf{p}_T}(t_0, t_f) + \mathbf{R}_{i,p} \right\|^2 + \left\| E_{\lambda^{[i]}}(t_0, t_f) - \angle \bar{\mathbf{p}}_{n,n}^{[i]'} \right\|^2, \tag{17}$$

with  $E_{\lambda^{[i]}}(t_0, t_f) = \angle \left( E_{\mathbf{p}_T}(t_0, t_f) - \mathbf{p}_{n,n}^{[i]} \right)$ , where we used the end point value property of BPs (Property A.1). Moreover, we notice that the cost function in Eq. (12) can be rewritten as

$$\int_{t_0}^{t_f} \left\| \angle \left( E_{\mathbf{p}_T}(t_0, \tau) - \mathbf{p}^{[i]}(\tau) \right) - \angle \dot{\mathbf{p}}^{[i]}(\tau) \right\|^2 d\tau.$$

Minimization of the above cost function is equivalent to minimizing

$$\int_{t_0}^{t_f} \eta^{[i]}(\tau) d\tau \triangleq \int_{t_0}^{t_f} \left\| \left( E_{p_T}(t_0, \tau) - \mathbf{p}^{[i]}(\tau) \right) \times \dot{\mathbf{p}}^{[i]}(\tau) \right\|^2 d\tau. \tag{18}$$

It is straightforward to show that the argument of the integral above is a Bernstein polynomial of order  $n$ , i.e.,  $\eta^{[i]}(t) = \sum_{j=0}^n \bar{\eta}_{j,n}^{[i]} b_{j,n}$ , and the integral reduces to the sum of its Bernstein coefficients by virtue of Property A.2, namely  $\int_{t_0}^{t_f} \eta^{[i]}(\tau) d\tau = \sum_{j=0}^n \bar{\eta}_{j,n}^{[i]}$ . The Bernstein coefficients  $\bar{\eta}_{j,n}^{[i]}$ ,  $j = 0, \dots, n$ , can be computed using Property A.5 of BPs.

The boundary conditions can be easily enforced by using the end point value property as follows:

$$\bar{\mathbf{p}}_{0,n}^{[i]} = \mathbf{p}_0^{[i]}, \quad \left\| \bar{\mathbf{p}}_{0,n}^{[i]'} \right\| = v_0^{[i]}, \quad \angle \bar{\mathbf{p}}_{0,n}^{[i]'} = \psi_0^{[i]}, \quad \bar{\omega}_{0,n}^{[i]} = \omega_0^{[i]}, \tag{19}$$

To impose the feasible and flyable constraints, we notice that the square of the speed of the UAV (see Eq. (15)) is a  $2n$ th order Bernstein polynomial, and its coefficients can be computed from the coefficients  $\bar{\mathbf{p}}^{[i]}$ . Thus, it is convenient to express the constraint in Eq. (7) as follows:

$$v_{\min}^2 \leq \|\dot{\mathbf{p}}_{i,n}(t)\|^2 \leq v_{\max}^2, \quad \forall t \in [t_0, t_f]. \tag{20}$$

The latter can be enforced using the *Evaluating Bounds* and *Evaluating Extrema* algorithms presented in [19]. Similarly, the angular rate constraint

$$\|\omega^{[i]}(t)\| \leq \omega_{\max}, \quad \forall t \in [t_0, t_f], \tag{21}$$

can be enforced using the same methods, since the angular rate is a rational BP (see Eq. (16) and Property A.5). The inter-vehicle safety constraint

$$\|\mathbf{p}^{[j]}(t) - \mathbf{p}^{[i]}(t)\| \geq D_s, \tag{22}$$

$\forall i, j = 1, \dots, m, i \neq j$ , and the no-fly-zone constraint

$$\|\mathbf{p}^{[i]}(t) - E_{p_T}(t_0, t)\| \geq R_T, \tag{23}$$

$\forall i = 1, \dots, m$ , can be efficiently imposed using the *Minimum Distance* and *Collision Detection* algorithms presented in [19].

Then, the *continuous-time*  $j$ th *flight* and *monitoring* trajectory generation problems for UAV <sub>$i$</sub>  (Problem 3.1) can be reformulated as finite dimensional optimization problems as follows:

**Flight Trajectory Generation** For given  $t_0$  and  $t_f = t_j^{[i]}$ , find  $\mathbf{p}_{j,n}^{[i]}$ ,  $j = 0, \dots, n$ , that minimize the cost function given by Eq. (17) subject to Eq0. (19)–(23).

Similarly, the (continuous-time)  $j$ th *monitoring* trajectory generation problem for UAV <sub>$i$</sub>  (Problem 3.2) can be reformulated as follows:

**Monitoring Trajectory Generation** For given  $t_0$  and  $t_f = t_j^{[i]} + T_\Delta$ , find  $\mathbf{p}_{j,n}^{[i]}$ ,  $j = 0, \dots, n$ , that minimize the cost function given by Eq. (18) subject to Eqs. (19)–(23).

**Remark 3.1** Due to the geometric properties of BPs (see Appendix and [7,19]), constraints (20)–(22) can be enforced for the whole trajectories (i.e.  $\forall t \in [t_0, t_f]$ ) by means of arithmetic operations on the Bernstein coefficients, independently of the order of the polynomials. This also implies that, to guarantee inter-vehicle safety, each vehicle must communicate only a finite (possibly small) number of coefficients to the rest of the fleet.

**Remark 3.2** The above reformulated flight and monitoring trajectory generation problems can be solved using an off-the-shelf optimization solver such as MATLAB's `fmincon` or `SNOPT`. The computation time required to solve is in the order of milliseconds, and this makes it feasible for real time implementation. This, in turn, guarantees robustness and safety of the mission even with unpredictable target movements.

## 4 Results

The BeBOT library [19] as well as Scipy's Optimize subpackage, Numpy, Matplotlib, and Numba are used to generate the following results. The tests were run on a Lenovo Thinkpad P52s laptop with 8 GB of RAM and an Intel Core i7-8550U processor running at 1.8 GHz. The implementation used to run the tests is available on GitHub at [https://github.com/caslabuoiowa/stochastic\\_target](https://github.com/caslabuoiowa/stochastic_target).

Consider a mission involving three UAVs. At  $t_1^{[1]} = 30$ s, UAV<sub>1</sub> is tasked with starting its monitoring trajectory. The prescribed monitoring maneuver mission is  $T_\Delta = 10$ s. Thus, we have  $\mathcal{I}^{[1]} = \{30, 60, 90, \dots\}$ s,  $\mathcal{I}^{[2]} = \{40, 70, 100, \dots\}$ s, and  $\mathcal{I}^{[3]} = \{50, 80, 110, \dots\}$ s. The minimum safe distance,  $D_s = 1$ m, outer monitoring radius,  $R_O = 75$ m, maximum speed,  $v_{\max} = 10 \frac{\text{m}}{\text{s}}$ , minimum speed,  $v_{\min} = 1 \frac{\text{m}}{\text{s}}$ , maximum angular rate,  $\omega_{\max} = 0.6 \frac{\text{rad}}{\text{s}}$ , and re-planning radius,  $\epsilon = 5$  m are selected. The UAVs are initially positioned at  $\mathbf{p}_0^{[1]} = [0, 0]^\top$  m,  $\mathbf{p}_0^{[2]} = [0, 25]^\top$  m, and  $\mathbf{p}_0^{[3]} = [0, 50]^\top$  m, respectively. They have initial headings  $\psi_0^{[1]} = \psi_0^{[2]} = \psi_0^{[3]} = 0$  rad, initial speeds  $v_0^{[1]} = v_0^{[2]} = v_0^{[3]} = 3 \frac{\text{m}}{\text{s}}$ , and initial angular rates  $\omega_0^{[1]} = \omega_0^{[2]} = \omega_0^{[3]} = 0 \frac{\text{rad}}{\text{s}}$ .

**Scenario 1** Consider a static target, with  $\mathbf{p}_T(t) = [25, 25]^\top$  m. The flight trajectories are computed at  $t_0 = 0$ , see Fig. 2a. The red dotted lines show the outer monitoring radii of the target at specified times. The gray circle in the center represents the no-fly-zone. The red star is the target and the solid colored lines are the agent flight trajectories. As the mission progresses, the agents generate and follow optimal monitoring trajectories (dashed lines), Fig. 2b. The previous flight trajectories are shown in light cyan, light green, and light blue. At  $t = 60$ s in Fig. 2b, Agent 1 is about to start its monitoring trajectory while Agent 3 is finishing its monitoring trajectory. On the other hand, Agent

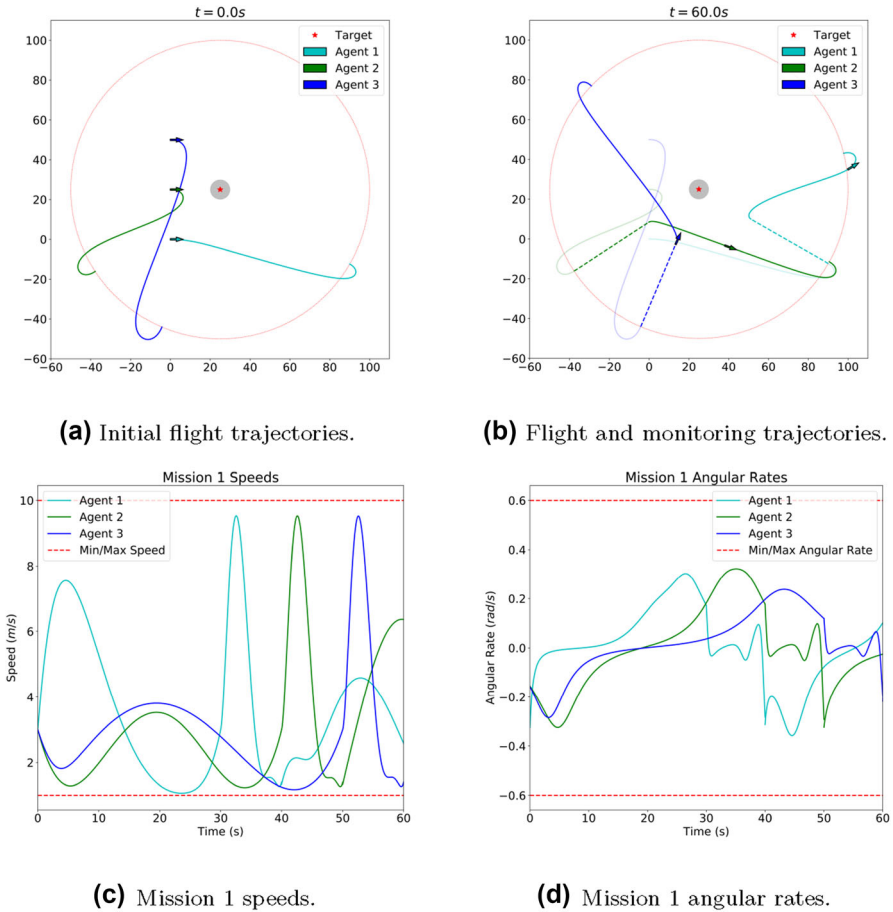


Fig. 2 Mission Scenario #1

2 is on its way to the outer radius and will begin its monitoring trajectory once Agent 1 completes its monitoring trajectory. Agent 2 had been monitoring the target before Agent 3. As expected, the monitoring trajectories all face radially inward towards the static target.

New flight trajectories are planned and bring the monitoring agents to a new, random position on the outer monitoring radius. The time history of the speeds and angular rates of the vehicles remain within the prescribed bounds, as shown in Fig. 2c, d.

**Scenario 2** Consider a target initially positioned at  $\mathbf{p}_T(0) = [25, 25]^T$  m, with initial heading angle of 0rad, initial speed of  $3 \frac{m}{s}$ , and initial angular rate of  $0 \frac{rad}{s}$ . At time  $t = 1s$ , the target starts moving with speed  $3 \frac{m}{s}, \forall t \geq 1s$ . It is assumed that the UAVs can only measure the target’s state once per second. Thus, it is not until  $t = 2s$  that the expected future position of the target is found to be outside of the radius  $\epsilon$ . As a result, re-planning is triggered, see Fig. 3. Note that the new trajectories still retain the same final time as the original flight trajectories, but were simply re-planned to

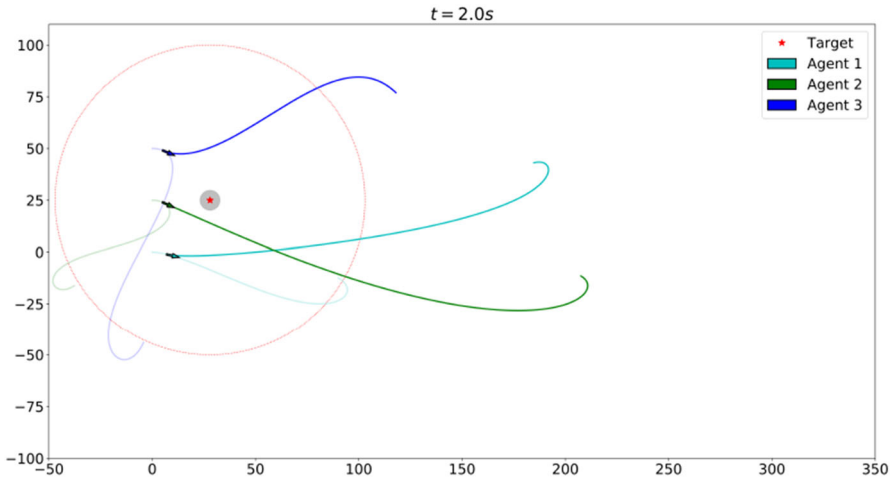
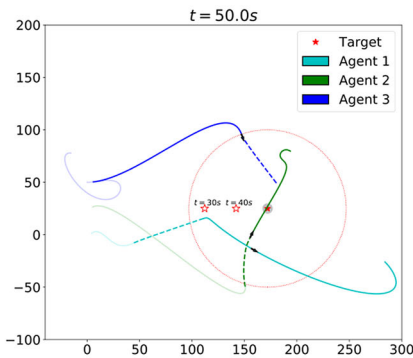
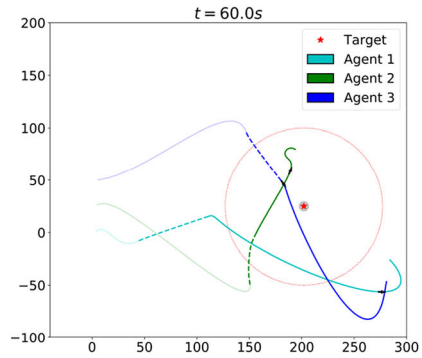


Fig. 3 Mission Scenario #2 for  $t = 2s$



(a) Mission 2 at  $t = 50s$ .

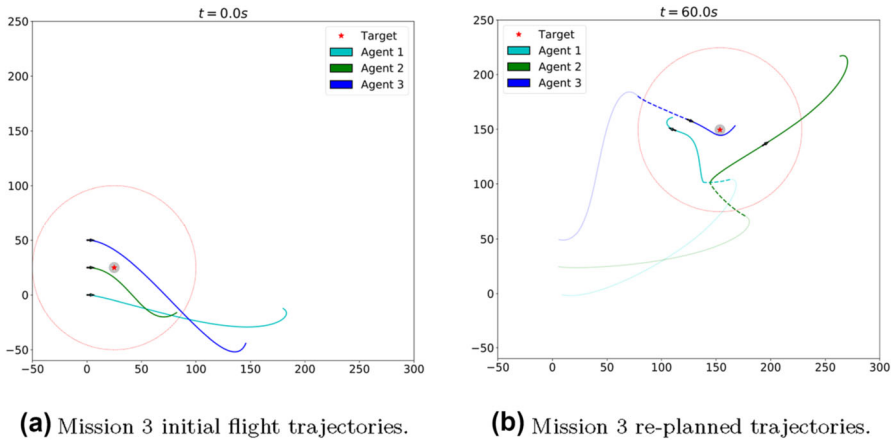


(b) Mission 2 at  $t = 60s$ .

Fig. 4 Mission Scenario #2 for  $t = 50s$  and  $t = 60s$

accommodate the new motion of the target. As the mission progresses, the agents continue to monitor the target. At time  $t = 50s$ , the monitoring trajectories track the position of the target, shown in Fig. 4a. Unfilled star symbols are used to show the past positions of the target at times  $t = 30s$  and  $t = 40s$ . The previously computed flight trajectories are shown in light cyan, light green, and light blue. At time  $t = 60s$ , Fig. 4b, UAV<sub>3</sub> finishes its monitoring trajectory.

**Scenario 3** The target is initially positioned at  $\mathbf{p}_T(0) = [25, 25]^T m$ , with initial heading angle of 0 radians, initial speed of  $3 \frac{m}{s}$ , and initial angular rate of  $0 \frac{rad}{s}$ . At time  $t = 1s$ , the target quickly changes its heading from 0 radians to  $\frac{\pi}{4}$  radians. Similarly to Scenario 2, the agents detect the change in the predicted future position of the target and re-plan their trajectories. At time  $t = 60s$  the agents can be seen to be following the new trajectory of the target, shown in Fig. 5b.



**Fig. 5** Mission Scenario #3

## 5 Conclusions

The problem of monitoring a stochastic target using a team of autonomous agents was formulated as a continuous time infinite dimensional optimal control problem. This problem was simplified using the sequential approach and then transcribed into a nonlinear optimization problem by restricting to the class of solutions defined by Bernstein polynomials. Two separate trajectories were computed for each agent: a flight trajectory and a monitoring trajectory. The flight trajectory aimed to safely navigate the agent to the outer monitoring radius and the monitoring trajectory optimized the line of sight with respect to the target position. To improve the success of monitoring the uncertain target, the trajectories were re-planned any time the predicted future position of the target deviated more than a radius of  $\epsilon$  from the original predicted position. The proposed method was demonstrated in simulation using three monitoring agents. The agents were able to safely reach the outer monitoring radius and then produced optimal monitoring trajectories. When the expected position of the target was predicted to lie outside the radius,  $\epsilon$ , of the original prediction, a new flight trajectory was generated to guarantee monitoring without lapses. Plots of these simulations were shown along with plots of the speed and angular rates to verify that constraints were satisfied.

While the examples shown only include a team of three vehicles, this method can be extended to a team with more or less vehicles. In the case for a team of two, one vehicle would be monitoring the trajectory while the second vehicle travels to its initial monitoring position. Two vehicles would provide a cheaper mission cost at the expense of less redundancy in monitoring. Similarly, a team of four vehicles could be employed to improve the chances of a successful mission and provide more time for each vehicle to reach the initial monitoring position. Depending on hardware being used, the number of monitoring vehicles can be increased even further. The interested reader is referred to [20], where numerical results involving multiple vehicle missions with larger number of agents are presented.

The direct method approach using Bernstein polynomials provides the user with an intuitive interface for including additional mission objectives or constraints by simply tweaking the cost and constraint functions. In addition to that, Bernstein polynomials provide computationally efficient properties and algorithms with which the cost and constraint functions can be computed. Due to the curse of dimensionality, this method does not scale to swarms of hundreds or more vehicles. However, the cost and feasibility of such a swarm for monitoring an individual target would likely limit this approach.

Future work includes incorporating stochastic dynamics of the target into the cost function. This will potentially be achieved by leveraging Wiener’s Polynomial Chaos [40] and Gaussian Process Regression (e.g. [39]) to express the future position of an unknown target in polynomial form, which can be easily incorporated into the problem formulation presented in this paper. We are also interested in developing methods to make the system more robust with respect to challenges such as an unreliable communication network, a target actively evading monitoring, and limited fuel availability.

**Acknowledgements** This research was supported by the Office of Naval Research, Grant N000142112091, with Ms. Christine Buzzell program officer.

## A Appendix

An  $n$ th order Bernstein polynomial,  $\mathbf{p}_n(t)$ , is defined as

$$\mathbf{p}_n(t) = \sum_{k=0}^n \bar{\mathbf{p}}_{k,n} b_{k,n}(t), \quad b_{k,n}(t) = \binom{n}{k} \frac{(t - t_0)^k (t_f - t)^{n-k}}{(t_f - t_0)^n}, \quad (24)$$

$t \in [t_0, t_f]$ , where  $\bar{\mathbf{p}}_{k,n} \in \mathbb{R}^N$  is the  $k$ th control point,  $b_{k,n}(t)$ ,  $k = 0, \dots, n$ , is the Bernstein basis, and  $\binom{n}{k}$  is the binomial coefficient. Bernstein polynomials (BPs) can be used to describe 2D (or 3D) spatial curves. In this case, BPs are often referred to as Bézier curves.

Furthermore, an  $n$ th order *rational* Bernstein polynomial,  $r_n(t)$  is defined as

$$r_n(t) = \frac{\sum_{k=0}^n \bar{\mathbf{p}}_{k,n} \bar{w}_{k,n} b_{k,n}(t)}{\sum_{k=0}^n \bar{w}_{k,n} b_{k,n}(t)}, \quad t \in [t_0, t_f], \quad (25)$$

where  $\bar{w}_{k,n} \in \mathbb{R}$ ,  $k = 0, \dots, n$  are referred to as weights.

Next, we provide a review of relevant properties of (rational) BPs, which are used in this paper. For an extensive review on BPs, the reader is referred to [9,17,19].

### Property A.1 End Point Values

The first and last control points of a (rational) Bernstein polynomial are its end-points, i.e.,  $\mathbf{p}_n(t_0) = \bar{\mathbf{p}}_{0,n}$  and  $\mathbf{p}_n(t_f) = \bar{\mathbf{p}}_{n,n}$ .

**Property A.2** *Integration*

The definite integral of a Bernstein polynomial is computed as

$$\int_{t_0}^{t_f} p_n(t)dt = w \sum_{k=0}^n \bar{p}_{k,n}, \quad w = \frac{t_f - t_0}{n + 1}. \tag{26}$$

**Property A.3** *Differentiation*

The derivative of a Bernstein polynomial is an  $(n - 1)$ th order Bernstein polynomial with vector of control points  $\bar{p}'_{n-1} = [\bar{p}'_{0,n-1}, \dots, \bar{p}'_{n-1,n-1}]$  given by  $\bar{p}'_{n-1} = \bar{p}_n \mathbf{D}_{n-1}$ , where  $\mathbf{D}_{n-1}$  is the  $\mathbb{R}^{(n-1) \times n}$  differentiation matrix (see [19]).

**Property A.4** *Degree Elevation*

Any Bernstein polynomial of degree  $n$  can be expressed as a Bernstein polynomial of degree  $m$ ,  $m > n$ . The vector of control points of the degree elevated Bernstein polynomial, namely  $\bar{p}_m = [\bar{p}_{0,m}, \dots, \bar{p}_{m,m}]$ , can be calculated as  $\bar{p}_m = \bar{p}_n \mathbf{E}_n^m$ , where  $\mathbf{E}_n^m = \{e_{j,k}\} \in \mathbb{R}^{(n+1) \times (m+1)}$  is the degree elevation matrix with elements given by  $e_{i,i+j} = \frac{\binom{m-n}{j} \binom{n}{i}}{\binom{m}{i+j}}$ , where  $i = 0, \dots, n$  and  $j = 0, \dots, m - n$ , all other values in the matrix are zero, and  $\bar{p}_n = [\bar{p}_{0,n}, \dots, \bar{p}_{n,n}]$  is the vector of control points of the curve being elevated (see [23]).

**Property A.5** *Arithmetic Operations*

The sum (difference) of two polynomials of the same order can be performed by simply adding (subtracting) their control points.

Let  $f_m(t)$  and  $g_n(t)$  be two 1-dimensional BPs of degree  $m$  and  $n$ , respectively, with control points  $\bar{a}_{0,m}, \dots, \bar{a}_{m,m}$  and  $\bar{b}_{0,n}, \dots, \bar{b}_{n,n}$ . The product  $h_{m+n}(t) = f_m(t)g_n(t)$  is a Bernstein polynomial of degree  $(m + n)$  with control points  $\bar{p}_{k,m+n}$ ,  $k \in \{0, \dots, m + n\}$  given by

$$\bar{p}_{k,m+n} = \sum_{j=\max(0,k-n)}^{\min(m,k)} \frac{\binom{m}{j} \binom{n}{k-j}}{\binom{m+n}{k}} \bar{a}_{j,m} \bar{b}_{k-j,n}. \tag{27}$$

The ratio between two 1-dimensional BPs,  $f_n(t)$  and  $g_n(t)$ , with control points  $\bar{a}_{0,n}, \dots, \bar{a}_{n,n}$  and  $\bar{b}_{0,n}, \dots, \bar{b}_{n,n}$ , i.e.,  $r_n(t) = f_n(t)/g_n(t)$ , can be expressed as a rational Bernstein polynomial as defined in (25), with control points and weights  $\bar{p}_{i,n} = \frac{\bar{a}_{i,n}}{\bar{b}_{i,n}}$ ,  $\bar{w}_{i,n} = \bar{b}_{i,n}$ , respectively.

**References**

1. Anderson, Ross P., Milutinović, D.: A stochastic approach to Dubins vehicle tracking problems. IEEE Trans. Automatic Control **59**(10), 2801–2806 (2014). <https://doi.org/10.1109/TAC.2014.2314224>
2. Betts, J.T.: Survey of numerical methods for trajectory optimization. J. Guid. Control. Dyn. **21**(2), 193–207 (1998). <https://doi.org/10.2514/2.4231>
3. Betts, J.T.: Practical methods for optimal control and estimation using nonlinear programming. SIAM (2010). <https://doi.org/10.1137/1.9780898718577>



4. Brown, S.S.: Optimal search for a moving target in discrete time and space. *Oper. Res.* **28**(6), 1275–1289 (1980). <https://doi.org/10.1287/opre.28.6.1275>
5. Casbeer, D.W., Kingston, D.B., Beard, R.W., McLain, T.W.: Cooperative forest fire surveillance using a team of small unmanned air vehicles. *Int. J. Syst. Sci.* **37**(6), 351–360 (2006). <https://doi.org/10.1080/00207720500438480>
6. Chung, H., Polak, E., Royset, J.O., Sastry, S.: On the optimal detection of an underwater intruder in a channel using unmanned underwater vehicles. *Naval Res. Logist. (NRL)* **58**(8), 804–820 (2011). <https://doi.org/10.1002/nav.20487>
7. Cichella, V., Kaminer, I., Dobrokhodov, V., Xargay, E., Hovakimyan, N., Pascoal, A.: Geometric 3D path-following control for a fixed-wing UAV on SO (3). In: *AIAA Guidance, Navigation, and Control Conference*, p. 6415 (2011). <https://doi.org/10.2514/6.2011-6415>
8. Cichella, V., Kaminer, I., Walton, C., Hovakimyan, N.: Optimal motion planning for differentially flat systems using Bernstein approximation. *IEEE Control Syst. Lett.* **2**(1), 181–186 (2017). <https://doi.org/10.1109/LCSYS.2017.2778313>
9. Cichella, V., Kaminer, I., Walton, C., Hovakimyan, N., Pascoal, A.: Bernstein approximation of optimal control problems. arXiv preprint [arXiv:1812.06132](https://arxiv.org/abs/1812.06132) (2018)
10. Cichella, V., Kaminer, I., Walton, C., Hovakimyan, N., Pascoal, A.M.: Consistent approximation of optimal control problems using Bernstein polynomials. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, pp. 4292–4297 (2019). <https://doi.org/10.1109/CDC40024.2019.9029677>
11. Cichella, V., Kaminer, I., Walton, C., Hovakimyan, N., Pascoal, A.M.: Optimal multi-vehicle motion planning using Bernstein approximants. *IEEE Trans. Autom. Control* (2020). <https://doi.org/10.1109/TAC.2020.2999329>
12. Conway, B.A.: A survey of methods available for the numerical optimization of continuous dynamic systems. *J. Optim. Theory Appl.* **152**(2), 271–306 (2012). <https://doi.org/10.1007/s10957-011-9918-z>
13. Ding, X.C., Rahmani, A.R., Egerstedt, M.: Multi-UAV convoy protection: an optimal approach to path planning and coordination. *IEEE Trans. Rob.* **26**(2), 256–268 (2010). <https://doi.org/10.1109/TRO.2010.2042325>
14. Dobbie, J.M.: Solution of some surveillance-evasion problems by the methods of differential games. In: *Proceedings of the 4th International Conference on Operational Research*, MIT, Wiley, New York (1966)
15. Dudek, G., Jenkin, M.R.M., Milios, E., Wilkes, D.: A taxonomy for multi-agent robotics. *Auton. Robot.* **3**(4), 375–397 (1996). <https://doi.org/10.1007/BF00240651>
16. Elijah, P.: Optimization: algorithms and consistent approximations (1997)
17. Farouki, R.T.: The Bernstein polynomial basis: a centennial retrospective. *Comput. Aided Geom. Des.* **29**(6), 379–419 (2012). <https://doi.org/10.1016/j.cagd.2012.03.001>
18. Kaminer, I., Pascoal, A.M., Xargay, E., Hovakimyan, N., Cichella, V., Dobrokhodov, V.: *Time-Critical Cooperative Control of Autonomous Air Vehicles*. Butterworth-Heinemann, Vladimir (2017)
19. Kielas-Jensen, C., Cichella, V.: BeBOT: Bernstein polynomial toolkit for trajectory generation. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 3288–3293 (2019). <https://doi.org/10.1109/IROS40897.2019.8967564>
20. Kielas-Jensen, C., Cichella, V.: Bernstein polynomial-based transcription method for solving optimal trajectory generation problems. arXiv preprint [arXiv:2010.09992](https://arxiv.org/abs/2010.09992) (2020)
21. Lalish, E., Morgansen, K.A., Tsukamaki, T.: Oscillatory control for constant-speed unicycle-type vehicles. In: *2007 46th IEEE Conference on Decision and Control*, IEEE, pp. 5246–5251 (2007). <https://doi.org/10.1109/CDC.2007.4434910>
22. LaValle, S.M., González-Banos, H.H., Becker, C., Latombe, J.-C.: Motion strategies for maintaining visibility of a moving target. In: *Proceedings of International Conference on Robotics and Automation*, vol. 1, IEEE, pp. 731–736 (1997). <https://doi.org/10.1109/ROBOT.1997.620122>
23. Lee, B.-G., Park, Y.: Distance for Bézier curves and degree reduction. *Bull. Aust. Math. Soc.* **56**(3), 507–515 (1997). <https://doi.org/10.1017/S0004972700031312>
24. Lee, J., Huang, R., Vaughn, A., Xiao, X., Hedrick, J.K., Zennaro, M., Sengupta, R.: Strategies of path-planning for a UAV to track a ground vehicle. In: *AINS Conference* (2003)
25. Li, Z., Hovakimyan, N., Dobrokhodov, V., Kaminer, I.: Vision-based target tracking and motion estimation using a small UAV. In: *49th IEEE Conference on Decision and Control (CDC)*, IEEE, pp. 2505–2510 (2010). <https://doi.org/10.1109/CDC.2010.5718149>

26. Livermore, R.A.: Optimal UAV path planning for tracking a moving ground vehicle with a gimbaled camera. Technical Report, Air Force Institute of Technology Wright-Patterson AFB OH Graduate School of Engineering and Management (2014)
27. Maki, T., Horimoto, H., Ishihara, T., Kofuji, K.: Autonomous tracking of sea turtles based on multibeam imaging sonar: toward robotic observation of marine life. *IFAC-Papers OnLine* **52**(21), 86–90 (2019). <https://doi.org/10.1016/j.ifacol.2019.12.288>
28. Milutinović, D., Casbeer, D.W., Kingston, D., Rasmussen, S.: A stochastic approach to small UAV feedback control for target tracking and blind spot avoidance. In: 2017 IEEE Conference on Control Technology and Applications (CCTA), pp. 1031–1037 (2017). <https://doi.org/10.1109/CCTA.2017.8062595>
29. Quintero, S.A.P., Hespanha, J.P.: Vision-based target tracking with a small UAV: optimization-based control strategies. *Control Eng. Practice* **32**, 28–42 (2014). <https://doi.org/10.1016/j.conengprac.2014.07.007>
30. Quintero, S.A.P., Ludkovski, M., Hespanha, J.P.: Stochastic optimal coordination of small UAVs for target tracking using regression-based dynamic programming. *J. Intell. Robot. Syst.* **82**(1), 135–162 (2016). <https://doi.org/10.1007/s10846-015-0270-7>
31. Rao, A.V.: A survey of numerical methods for optimal control. *Adv. Astronaut. Sci.* **135**(1), 497–528 (2009)
32. Ren, W., Beard, R.W.: CLF-based tracking control for UAV kinematic models with saturation constraints. In: 42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475), vol. 4, IEEE, pp. 3924–3929 (2003). <https://doi.org/10.1109/CDC.2003.1271763>
33. Ross, I.M., Karpenko, M.: A review of pseudospectral optimal control: from theory to flight. *Annu. Rev. Control.* **36**(2), 182–197 (2012). <https://doi.org/10.1016/j.arcontrol.2012.09.002>
34. Royset, J.O.: Route optimization for multiple searchers. *Naval Res. Logist. (NRL)* **57**(8), 701–717 (2010). <https://doi.org/10.1002/nav.20432>
35. Schwartz, A., Polak, E.: Consistent approximations for optimal control problems based on Runge–Kutta integration. *SIAM J. Control. Optim.* **34**(4), 1235–1269 (1996). <https://doi.org/10.1137/S0363012994267352>
36. Schwartz, A.L.: Theory and implementation of numerical methods based on Runge–Kutta integration for solving optimal control problems. PhD Thesis, University of California, Berkeley (1996)
37. Stephens, S., Manyam, S.G., Casbeer, D.W., Cichella, V., Kunz, D.L.: Randomized continuous monitoring of a target by agents with turn radius constraints. In: 2019 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, pp. 588–595 (2019). <https://doi.org/10.1109/ICUAS.2019.8798373>
38. Sun, J., Li, B., Jiang, Y., Wen, C.: A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes. *Sensors* **16**(11), 1778 (2016). <https://doi.org/10.3390/s16111778>
39. Wahlström, N., Özkan, E.: Extended target tracking using Gaussian processes. *IEEE Trans. Signal Process.* **63**(16), 4165–4178 (2015). <https://doi.org/10.1109/TSP.2015.2424194>
40. Wiener, N.: The homogeneous chaos. *Am. J. Math.* **60**(4), 897–936 (1938). <https://doi.org/10.2307/2371268>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.