JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 31, No. 6, November–December 2008

Framework for Low-Observable Trajectory Generation in Presence of Multiple Radars

Tamer Inanc* and Mehmet K. Muezzinoglu*
University of Louisville, Louisville, Kentucky 40292
Kathleen Misovec†
BAE Systems, Burlington, Massachusetts 01803
and
Richard M. Murray‡
California Institute of Technology, Pasadena, California 91125

DOI: 10.2514/1.35287

This paper explores the problem of finding a real-time optimal trajectory for unmanned aerial vehicles to minimize their probability of detection by opponent multiple radar detection systems. The problem is handled using the nonlinear trajectory generation method developed by Milam et al. (Milam, M., Mushambi, K., and Murray, R., "New Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems," *Proceedings of the 39th IEEE Conference on Decision and Control*, Vol. 1, Institute of Electrical and Electronics Engineers, New York, Dec. 2000, pp. 845–851.) The paper presents a formulation of the trajectory generation task as an optimal control problem, where temporal constraints allow periods of high observability interspersed with periods of low observability. This feature can be used strategically to aid in avoiding detection by an opponent radar. The guidance is provided in the form of sampled tabular data. It is then shown that the success of nonlinear trajectory generation on the proposed low-observable trajectory generation problem depends upon an accurate parameterization of the guidance data. In particular, such an approximator is desired to have a compact architecture, a minimum number of design parameters, and a smooth continuously differentiable input–output mapping. Artificial neural networks as universal approximators are known to possess these features, and thus are considered here as appropriate candidates for this task. Comparison of artificial neural networks against B-spline approximators is provided, as well. Numerical simulations on multiple radar scenarios illustrate unmanned air vehicle trajectories optimized for both detectability and time.

Nomenclature

$B_{j,k_i}(t)$ = B-spline basis function for the output z_i
C^i_j = coefficients of the B-splines
e_{ac} = aircraft positions along the east axis
k_i = degree of spline polynomial
l_i = number of knot intervals
m_i = number of smoothness conditions at knot points
n_{ac} = aircraft positions along the north axis
p_i = number of coefficients of each output
r = slant range
$s(\varepsilon, \alpha)$ = signature function
u_{ac} = aircraft positions along the up axis
$\mathbf{u}(t)$ = control input
$\mathbf{x}(t)$ = state of the system
z = differentially flat output
α = azimuth angle
ε = elevation angle
ξ = heading angle

I. Introduction

UNMANNED aerial vehicles (UAVs) are remotely controlled (semi) autonomous aerial vehicles that can carry different sensors, communications equipments, and other payloads. UAVs have been used extensively for surveillance, intelligence gathering, reconnaissance, targeting, and target damage assessment, to "see over the next hill" missions without risking the lives of aircrew.

One of the main threats for UAVs is radar detection systems. Optimal trajectories for UAVs can reduce their detectability against radar systems. In [1], we use attitude-dependent signature models in conjunction with the nonlinear trajectory generation (NTG) method to find low-observable routes. We also develop temporal observability constraints for strategic detection avoidance. In [2], radar cross-section models are used to find paths with distinct periods for optimizing detection or time. Kim and Hespanha [3] developed anisotropic shortest-path methods for designing minimum risk flight paths. In [4], the feasibility of using geometric, deterministic solutions to optimal trajectories between two radars is explored. Work by McFarland et al. [5] uses motion planning techniques, using potential field theory for unmanned air vehicle path planning in the presence of detection systems. An analytical solution yielding the trajectory that minimizes the radar energy reflected from the target is derived in [6] using the calculus of variations.

Real-world threats can be much more complex than those modeled in prior trajectory generation work, which has tended to model threats as physical obstacles. In certain instances involving areas that are dense with threats, for example, path planning that relies primarily on obstacle avoidance may rule out the simultaneous goals of traveling to a destination waypoint in a high-risk area and avoiding the threats. On the other hand, by modeling and managing the observability of the paths, path planners can be designed that have the potential to achieve these objectives.

Received 24 October 2007; revision received 15 January 2008; accepted for publication 7 February 2008.

*Assistant Professor, Electrical and Computer Engineering Department.

†Advanced Information Technologies Division; currently at Science, Engineering, & Technology Corporation.

‡Thomas E. and Doris Everhart Professor, Control and Dynamical Systems Department.

In this paper, we present a framework to the problem of finding low-observable, optimal flight paths in the presence of multiple radars. In contrast to an obstacle avoidance approach, our aim is to find a real-time nonlinear trajectory *maximizing* the probability of not-being-detected (PND) function, while minimizing the total flight time between a given base station and a final destination, thus minimizing the risk of the aircraft.

First, we present a simplified aircraft model, as well as the probability of detection and signature models that induce the guidance on the UAVs. These models are provided by the Boeing open experimental platform (OEP) [7] as tabular data. The detectability of an aircraft traveling near an opponent radar depends on more than just the distance to the radar; it depends on the aircraft attitude[§] and configuration as well. This feature in the threat model introduces nonconvexities, path dependencies, as well as sharp gradients into the underlying optimization problem, presenting new challenges for trajectory generation techniques. The models have a number of sharp gradients with low-detectability regions for both cone-shaped areas centered around the nose as well as around the rear of the UAV.

To apply any deterministic control method to trajectory generation problems in general, the designer requires an accurate identification of the guidance model on the state space, which might be available only in the form of finite measurements/observations. Building an approximate model of this essential component of the problem can be achieved by an auxiliary parametric approximator, in addition to the approximator integrated into the solver. This model would then take part in the differential equations governing the motion and/or the optimal control formulation.

Therefore, we next find analytical models to represent the guidance space, i.e., probability of detection and signature data, in the presence of multiple radar systems. These can be developed using any number of function approximation techniques. We use artificial neural networks (ANNs) and compare their results against tensor product B-spline functions [8].

It is natural to refer to B-splines for this external approximation task, as was done in [9]. Piecewise polynomials posses desirable features as static approximators, such as being easily differentiable and avoiding global dependence of the approximation on parameters, i.e., polynomial coefficients. On the other hand, their design may involve a massive trial-and-error process, because many critical assumptions, including the locations of collocation points, breakpoints, and the degrees of polynomials, have to be made before the computational phase. Optimization of piecewise polynomials is also constrained by continuity conditions imposed on the breakpoints. In short, despite some commonsense and empirically derived conditions for choosing these parameters, it is a fact that B-spline approximation may be too costly a procedure to be applied in real-time control tasks.

ANNs have been established as universal approximators [10] due to their effective and tractable training methods [11]. The assumptions on the neural network architecture before training are minimal, and so their design phase is mostly computational rather than structural. Moreover, the decision on the network architecture may be formulated and embedded in the parameter optimization phase, eliminating data-dependent trial-and-error efforts [12] and the risk of overfitting the data. The training mechanism is a simple unconstrained gradient-descent procedure, not dealing with any conditions on the continuity of the network transfer function. In addition to these advantages in the design phase, ANNs are among the cheapest algebraic architectures in terms of computation cost, as they simply realize a weighted sum of their inputs and pass the outcome through an algebraic nonlinearity. Depending on the choice of this nonlinearity, it is also possible to avoid each parameter of the network from affecting the entire mapping, providing almost all desired features of B-splines in the approximation task previously defined.

Next, we define an objective function that is a weighted combination of time, speed, and probability of detection. The constraint functions include initial and final conditions, speed constraints, turning radius constraints, as well as temporal constraints. Temporal constraints [13,14] are used to allow limited time periods of high-risk flight [1].

The paper is organized as follows. The next section gives the necessary background on NTG and tensor product B-splines. Section III introduces aircraft and detection models, and formulation of the trajectory generation problem. The use of ANNs and B-splines to express the guidance to be processed by NTG is proposed in Sec. IV. In Sec. V, we present simulations with multiple radars. These examples illustrate how real-time nonlinear trajectories are found to optimize an objective function of a weighted sum of the probability of detection, speed, and total mission time of a UAV without modeling radars as obstacles. Concluding remarks are given in the last section.

II. Background

In this section, a brief introduction to the nonlinear trajectory generation algorithm is given.

A. Nonlinear Trajectory Generation Algorithm

The nonlinear trajectory generation algorithm developed by Milam [15] solves constrained nonlinear optimal control problems in real time. The main advantage of NTG compared with other dynamic optimization methods is that it can quickly provide suboptimal solutions, which makes it very useful for real-time applications. In addition, linear as well as nonlinear constraints and cost functions can be included in the problem formulation of NTG. The general NTG framework can handle both spatial and temporal constraints.

NTG is based on a combination of nonlinear control theory, spline theory, and sequential quadratic programming. NTG takes the optimal control problem formulation, characterization of trajectory space, and the set of collocation points, and transforms them into a nonlinear programming (NLP) problem. The transformed NLP problem is then solved using NPSOL [16], a popular NLP solver, which uses sequential quadratic programming (SQP). There are three steps in the NTG algorithm.

1. *Parameterization*

The first step is to exploit any differential flatness[¶] of the given system to find a new set of outputs, so that the system dynamics can be mapped down to a lower-dimensional space where all the states and controls of the original system can be recovered from the new lower-dimensional representation [17].

Consider a general dynamic (control) system

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \tag{1}$$

For optimal control, we would like to choose $\mathbf{u}(t)$ such that some cost function is minimized and constraints are enforced. That is, given a cost function of the form

$$J = \Phi_0(\mathbf{x}(t_0), \mathbf{u}(t_0), t_0) + \int_{t_0}^{t_f} \Phi_t(\mathbf{x}(t), \mathbf{u}(t), t)\,\mathrm{d}t + \Phi_f(\mathbf{x}(t_f), \mathbf{u}(t_f), t_f) \tag{2}$$

we would like to choose $\mathbf{u}(t)$ for $t \in [t_0, t_f]$ which minimizes J subject to constraints of the form

$$\begin{aligned} &\text{Initial } lb_0 \le \Psi_0(\mathbf{x}(t_0), \mathbf{u}(t_0), t_0) \le ub_0 \\ &\text{Trajectory } lb_t \le \Psi_t(\mathbf{x}(t), \mathbf{u}(t), t) \le ub_t \\ &\text{Final } lb_f \le \Psi_f(\mathbf{x}(t_f), \mathbf{u}(t_f), t_f) \le ub_f \end{aligned} \tag{3}$$

[§] The aircraft attitude affects detectability because it affects the viewing aspect angles presented to the radar.

[¶] A system is differentially flat if a set of outputs exists such that all states and inputs can be composed from these outputs (and their derivatives) without integration.

Notice that the cost function J is composed of an initial condition cost $\Phi_0(\cdot)$, an integral cost over the trajectory $\Phi_t(\cdot)$, and a final condition cost $\Phi_f(\cdot)$. The constraints are similarly partitioned as Ψ_0,Ψ_t, and Ψ_f represent initial, trajectory, and final constraints, respectively. Cost and constraint functions might be linear or nonlinear functions. Lower and upper bounds are represented by lb and ub, respectively. Equations (2) and (3) are standard in optimal control and are further explained in [15,18].

The system in Eq. (1) can be mapped to a lower-dimensional space, where it will be easier and computationally more efficient to solve the optimization problem, by finding an output $z = z_1, \ldots, z_q$ of the form

$$z = A(x, u, u^{(1)}, \ldots, u^{(r)}) \tag{4}$$

with $u^{(i)}$ denoting the ith derivative of u with respect to time.

If Eq. (1) is differentially flat, then the states and inputs of the system (x, u) can be completely recovered from Eq. (5):

$$(x, u) = B(z, z^{(1)}, \ldots, z^{(s)}) \tag{5}$$

Here, $z^{(i)}$ denotes the ith derivative of z with respect to time. A necessary condition for the existence of such outputs is given in [19]. Special cases in which no flat output exists can be handled as described in [20].

2. *B-Spline Representation*

In most cases, the dynamics in Eq. (1) and constraints in Eq. (3) are too complicated for the minimization of Eq. (2) to be solved analytically, and so numerical solutions are required. To solve optimal control problems numerically, they are often transformed into nonlinear programming problems. To achieve this transformation, the NTG algorithm further represents the outputs, given in Eq. (4), in terms of the B-spline functions [15,20] as

$$z_i(t) = \sum_{j=1}^{p_i} B_{j,k_i}(t) C_j^i \tag{6}$$

for the knot sequence t_i, $i = 1, \ldots, q$ and where $p_i = l_i(k_i - m_i) + m_i$. Therefore, the output set $(z_1, z_2, \ldots, z_q)$ is represented by

$$P = \sum_{i=1}^{q} p_i$$

coefficients. Figure 1 shows the B-spline representation of outputs. This B-spline parameterization constitutes the *internal approximator* of the NTG framework.

3. *Reformulation*

The final step of NTG is to reformulate the cost and constraint functions given in Eqs. (2) and (3) in terms of the B-spline coefficients C_j^i to be solved by the sequential quadratic programming package NPSOL. This reformulation yields a nonlinear programming problem. It is necessary to discretize the time interval $[t_0, t_f]$ into w intervals with $w + 1$ breakpoints to translate the problem into the SQP framework. In general, $w + 1$ breakpoints, where the trajectory constraints will be satisfied, are chosen uniformly over the time interval $[t_0, t_f]$, though nonuniform or Gaussian distribution may also be considered. The optimization problem, defined by Eqs. (2) and (3), then can be stated in the following NLP form:

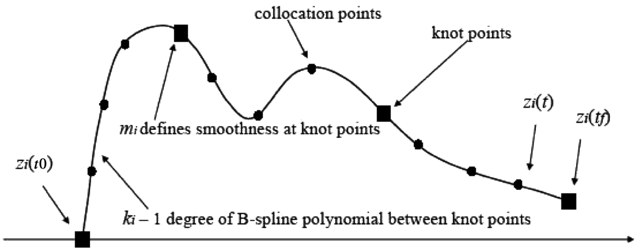

collocation points
knot points
zi(t)
zi(tf)
mi defines smoothness at knot points
zi(t0)
ki – 1 degree of B-spline polynomial between knot points

Fig. 1 B-spline representation of outputs.

$$\min_{\mathbf{C}\in\mathbb{R}^P} F(\mathbf{C}) \quad \text{subject to } L \le G(\mathbf{C}) \le U \tag{7}$$

where $\mathbf{C} = [C_1^1, \ldots, C_{p_1}^1, C_1^2, \ldots, C_{p_2}^2, \ldots, C_1^q, \ldots, C_{p_q}^q]^T$ and

$$P = \sum_{i=1}^{q} p_i$$

The transformed cost function is $F(\mathbf{C})$, and $G(\mathbf{C})$ is the transformation of the constraints, with L and U being the lower and upper bounds, respectively. The remainder of the solution is then a matter of nonlinear programming involving coefficients.

Finally, the user specifies a problem to NTG by stating the problem in terms of some choice of outputs and their derivatives, providing the cost and the constraints in terms of these outputs and their derivatives, specifying the regularity of the variables, the placement of the knot points, the order and regularity of the B-splines, and the breakpoints for each output.

B. Using Temporal Constraints with Nonlinear Trajectory Generation

Although the NTG formulation allows any spatial constraint to be easily coded into the constraint set, including temporal constraints requires more care. The easiest way to solve this is to introduce time as a state variable in the optimization [21,22].

First, define the new scaled time variable τ shown in Eq. (8), where t represents the true time, or old time, and T is the new state variable representing the unknown final time which will be optimized. In the setup of the optimization problem detailed in Sec. III.C, scaled time τ rather than true time t is used:

$$\tau = \frac{t}{T} \tag{8}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \to \frac{\mathrm{d}}{\mathrm{d}\tau}\frac{\mathrm{d}\tau}{\mathrm{d}t} = \frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}\tau} \tag{9}$$

After introducing the new state variable T, the cost and constraint functions given in Eqs. (2) and (3) become

$$\begin{aligned} J(\mathbf{x}, \mathbf{u}, T) &= \Phi_0(\mathbf{x}(0), \mathbf{u}(0), T) \\ &\quad + \Phi_f(\mathbf{x}(1), \mathbf{u}(1), T) + \int_0^1 \Phi_t(\mathbf{x}(\tau), \mathbf{u}(\tau), T)\,\mathrm{d}\tau \end{aligned} \tag{10}$$

$$\begin{aligned} lb_0 &\le \Psi_0(\mathbf{x}(0), \mathbf{u}(0), T) \le ub_0 \\ lb_f &\le \Psi_f(\mathbf{x}(1), \mathbf{u}(1), T) \le ub_f \\ lb_t &\le \Psi_t(\mathbf{x}(\tau), \mathbf{u}(\tau), T) \le ub_t \end{aligned} \tag{11}$$

Note that without loss of generality, it is assumed that $t_0 = 0$ and $t_f = T$. Therefore, as $t \to 0$, $\tau \to 0$ and $t \to T$, $\tau \to 1$. Any additional temporal constraints may be expressed as a set of inequalities given by

$$lb_T \le \Psi_T(T) \le ub_T \qquad N_T \text{ temporal constraints}$$

III. Low-Observable Trajectory Generation

Now that we have outlined the general methods, we focus on how to apply these methods to the low-observable trajectory generation problem.

A. Aircraft Model

The aircraft and the detection models, as shown in Fig. 2, are the two main components based on the open experimental platform

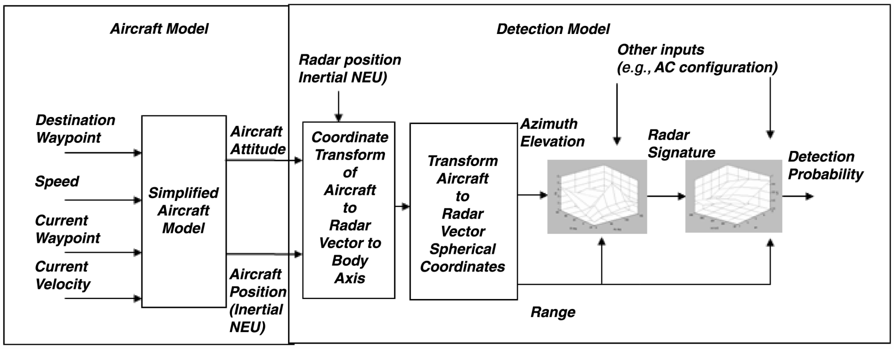

Aircraft Model
Detection Model
Radar position Inertial NEU)
Other inputs (e.g., AC configuration)
Destination Waypoint
Speed
Current Waypoint
Current Velocity
Simplified Aircraft Model
Aircraft Attitude
Aircraft Position (Inertial NEU)
Coordinate Transform of Aircraft to Radar Vector to Body Axis
Transform Aircraft to Radar Vector Spherical Coordinates
Azimuth Elevation
Radar Signature
Detection Probability
Range

Fig. 2 Aircraft and detection models.

developed by Boeing [7] as part of the Defense Advanced Research Projects Agency–Mixed Initiative Control of Automa-teams (DARPA-MICA) program.** The simplified two-dimensional (2D) version of the OEP aircraft model used here is the same as in [1]. However, the OEP also models the relationship between bank angle and turn rate. Modeling these additional dynamics is useful for achieving further reductions in the probability of detection. For example, when the aircraft dives, climbs, or banks, the signature and probability of detection are affected. NTG can be extended to handle these effects and work is progressing in this direction. We assume here that 1) the UAV maintains a constant altitude and the pitch angle is zero, and 2) the bank angle is zero. The first assumption implies that the azimuth effects are more dominant than the elevation effects. This assumption is more realistic at standoff ranges because elevation has a more pronounced effect as the aircraft gets closer in distance to the radar. The second assumption is derived from assuming that the turn times are small compared with the time between waypoints. This assumption becomes less valid as the waypoint routes become finer.

Current position, velocity, heading, and the destination position are the inputs to the aircraft model, whereas the outputs are the aircraft position and attitude in inertial coordinates (north, east, up) as defined in the OEP model. The following point mass model equations represent the state equations for the vehicle traveling from waypoint i to the next waypoint $i+1$:

$$\dot{n}_{\mathrm{ac}} = V_i \cos(\xi_i) \qquad \dot{e}_{\mathrm{ac}} = V_i \sin(\xi_i) \qquad \dot{u}_{\mathrm{ac}} = 0 \qquad \xi_i = \tan^{-1}\frac{\dot{e}_{\mathrm{ac}}}{\dot{n}_{\mathrm{ac}}} \tag{12}$$

V is the speed and ξ is the heading, which is the angle between the nose and north, and is positive clockwise about the up axis. Assuming zero pitch and bank angles, a vector from the aircraft $(n_{\mathrm{ac}}, e_{\mathrm{ac}}, u_{\mathrm{ac}})$ to the radar (n_R, e_R, u_R) is given in body axes as

$$\begin{bmatrix} x_{R\mathrm{ac}} \\ y_{R\mathrm{ac}} \\ z_{R\mathrm{ac}} \end{bmatrix} = \begin{bmatrix} \cos(\xi) & \sin(\xi) & 0 \\ -\sin(\xi) & \cos(\xi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (n_R - n_{\mathrm{ac}}) \\ (e_R - e_{\mathrm{ac}}) \\ (u_R - u_{\mathrm{ac}}) \end{bmatrix} \tag{13}$$

B. Detection Model

Three scalars, namely elevation ε, azimuth α, and slant range r, computed using the inertial coordinates of UAV and the radar, are of special interest in the problem formulation. To obtain these values, the vector in Eq. (13) is transformed to spherical coordinates. For each radar, they are defined as follows:

$$\begin{aligned} \alpha &= \tan^{-1}\left(\frac{-(n_R - n_{\mathrm{ac}})\dot{e}_{\mathrm{ac}} + (e_R - e_{\mathrm{ac}})\dot{n}_{\mathrm{ac}}}{(n_R - n_{\mathrm{ac}})\dot{n}_{\mathrm{ac}} + (e_R - e_{\mathrm{ac}})\dot{e}_{\mathrm{ac}}}\right) \\ \varepsilon &= \tan^{-1}\left(\frac{u_R - u_{\mathrm{ac}}}{\sqrt{(n_R - n_{\mathrm{ac}})^2 + (e_R - e_{\mathrm{ac}})^2}}\right) \\ r &= \sqrt{(n_R - n_{\mathrm{ac}})^2 + (e_R - e_{\mathrm{ac}})^2 + (u_R - u_{\mathrm{ac}})^2} \end{aligned} \tag{14}$$

where the coordinates with subscript R are associated with the considered radar. Please note that for all $\tan^{-1}$ functions, the four quadrant inverse tangent functions were used.

The first two variables determine the signature value s_R of the UAV with respect to the associated radar R. This value is unitless and is related to the radar cross section. Its dependency on α and ε are not given in closed form, but sampled on representative combinations, as given in Table 1. Because the radars in our problem are assumed identical, the signature functions with respect to each radar are the same, i.e., $s_R = s$. Depending on the range r and the signature s of the UAV, with respect to each radar, there is an instantaneous probability of detection $\mathrm{PD}(s, r)$.

In our problem, this crucial component is not given as an analytical function of signature and range, but as samples at representative points, as given in Table 2. Note that each entry in this table is the range r (kilometers), where columns and rows index the signature value and the probability of detection, respectively.

Generalization of the detection probability from the samples given in Tables 1 and 2 will be necessary and sufficient to embed the detection model in the problem formulation as described next. The tables depend on the type of aircraft, the configuration of the aircraft, as well as the type of radar. Details of the aircraft dynamics and the detection model can be found in [7].

C. Problem Formulation

The trajectory generation problem on the previously described domain is a dynamic optimization problem cast as minimizing a weighted function of total mission time, velocity, and probability of detection. The goal is to determine the shortest path (minimizing the energy) from the start point that encounters each waypoint, while consuming minimum energy and maintaining minimal observability, i.e., minimizing the probability of detection by all radars. The given information includes radar locations and types, an aircraft type and configuration, a start location, initial heading angle, and a destination location of the aircraft. We also assume that radars are monostatic, containing a collocated transmitter and receiver, and each station uses a different frequency so that there is no interaction between observers.

The low-observability routing problem is considered here to be temporal in nature, by allowing periods of high observability interspersed with periods of low observability. As in [1], we define a set of two types of temporal events: 1) probability of not-being-detected function (PND) *must be* high between $T_{2i} \le t < T_{2i+1}$, and 2) probability of PND *could be* low between $T_{2i+1} \le t < T_{2i+2}$ for $i = 0, 1, \ldots, n$. High-PND times, where the UAV's observability to the radar systems is minimized, are scattered with the short periods of

**The Mixed Initiative Control of Automa-teams of DARPA studies a multilayer planning, assessment, and control architecture of distributed semi-autonomous forces with collective objectives.

Table 1 Signature table of considered UAV with respect to each radar

ε/α	0 deg	±30 deg	±31 deg	±159 deg	±160 deg	±180 deg
+90 deg	$1E0$	$1E0$	$1E0$	$1E0$	$1E0$	$1E0$
+45 deg	$5E-3$	$5E-3$	$1E0$	$1E0$	$5E-3$	$5E-3$
+20 deg	$5E-4$	$5E-4$	$5E-1$	$5E-1$	$5E-4$	$5E-4$
0 deg	$5E-5$	$5E-5$	$5E-1$	$5E-1$	$5E-5$	$5E-5$
−20 deg	$5E-4$	$5E-4$	$5E-1$	$5E-1$	$5E-4$	$5E-4$
−45 deg	$5E-3$	$5E-3$	$1E0$	$1E0$	$5E-3$	$5E-3$
−90 deg	$1E0$	$1E0$	$1E0$	$1E0$	$1E0$	$1E0$

Table 2 Relation between probability of detection and signature s via slant range r

s	PD = 0.99	PD = 0.50	PD = 0.10	PD = 0.01
1.0	275.0	348.2	402.1	475.2
$1E-1$	154.6	195.8	226.1	267.2
$1E-2$	87.0	110.1	127.2	150.3
$1E-3$	48.9	61.9	71.5	84.5
$1E-4$	27.5	34.8	40.2	47.5
$1E-5$	15.5	19.6	22.6	26.7
$1E-6$	8.7	11.0	12.7	15.0

low-PND times (high-observability regions). Such a relaxation in constraints can be useful when it is difficult to satisfy all competing objectives of the problem and allowing limited high-risk periods is acceptable. Figure 3 shows a simple illustration of this approach. The long, dashed lines represent the high-PND periods where the aircraft's observability is minimized, whereas the short solid lines represent the low-PND regions where short periods of high observability are allowed. In [1], this type of situation is discussed in more detail.

The unknown final event time is represented by T, which is introduced to be able to include temporal constraints in the NTG algorithm. Thus, in the following optimization formulations, the scaled time variable $\tau = t/T$ is used [21]. The system dynamics for this problem consist of the vehicle dynamics in Eq. (12), together with the following dynamics on the new state variables

$$\frac{\mathrm{d}T_{2i}}{\mathrm{d}\tau} = 0 \quad \text{and} \quad \frac{\mathrm{d}T_{2i+1}}{\mathrm{d}\tau} = 0 \qquad i = 0, \ldots, n \tag{15}$$

Note that the system is differentially flat. All variables of interest can be written as a function of the output variables n_{ac}, e_{ac}, T_{2i}, T_{2i+1}, and their derivatives.

Next, we define the PND function as

$$\mathrm{PND} = \prod_{k=1}^{N_r} (1 - \mathrm{PD}[k]) \tag{16}$$

where N_r represents number of radar systems. The probability of detection $\mathrm{PD} = f_{\mathrm{PD}}(s, r)$ is considered as a function of signature s and range r values, where $s = f_s(\varepsilon, \alpha)$ is a function of elevation ε and azimuth α angles. These two functions are constructed using ANNs, as well as using tensor product B-spline functions to fit analytical models to the signature and probability of detection data given in Tables 1 and 2, and it will be explained in more detail.

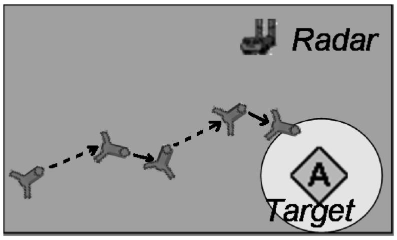


Fig. 3 Two temporal event types: high- and low-PND periods.

D. Cost Functions

Cost functions are given based on the programming structure of the NTG algorithm as follows:

1) Initial cost function

$$\Phi_0 = W_t T, \quad \text{where } T = \sum_{i=0}^{n} (T_{2i} + T_{2i+1})$$

Φ_0 serves for the purpose of minimizing the total mission time T only, W_t being the weight of this cost component.

2) Trajectory cost function

$$\int_0^1 \left\{ \frac{W_u}{T^2} \left[\left(\frac{\mathrm{d}n_{\mathrm{ac}}}{\mathrm{d}\tau} \right)^2 + \left(\frac{\mathrm{d}e_{\mathrm{ac}}}{\mathrm{d}\tau} \right)^2 \right] - W_p \mathrm{PND} \right\} T \, \mathrm{d}\tau$$

is defined along the scaled time interval τ, [0,1]. Positive coefficients W_u and W_p represent the weight functions on the speed and PND functions, respectively. The trajectory cost function bounds the control action by penalizing the speed and minimizing the observability of the aircraft by maximizing the PND function, which is always positive. Note the negative sign in front of the weight function W_p.

In the NTG code setup, the initial cost function is used to add constant terms to the cost, and the trajectory cost function subroutine is used for the costs that are integrated.

E. Constraint Functions

Based on the programming structure of the NTG algorithm, a set of constraint functions is given as

1) Linear initial constraints

$$n_{\mathrm{ac}}(\tau)|_{\tau=0} = n_{\mathrm{ac}}(0) \qquad e_{\mathrm{ac}}(\tau)|_{\tau=0} = e_{\mathrm{ac}}(0)$$

$$\left.\frac{\mathrm{d}n_{\mathrm{ac}}}{\mathrm{d}\tau}\right|_{\tau=0} = \dot{n}_{\mathrm{ac}}(0)T \qquad \left.\frac{\mathrm{d}e_{\mathrm{ac}}}{\mathrm{d}\tau}\right|_{\tau=0} = \dot{n}_{\mathrm{ac}}(0)T$$

$$\underline{T_{2i}} \le T_{2i}|_{\tau=0} \le \overline{T_{2i}}, \qquad i = 0, 1, \ldots, n$$

$$\underline{T_{2i+1}} \le T_{2i+1}|_{\tau=0} \le \overline{T_{2i+1}}, \qquad i = 0, 1, \ldots, n$$

where

$$\dot{(\cdot)} = \frac{d(\cdot)}{dt} \cdot \overline{T_{2i+1}} \ll \overline{T_{2i}}$$

where $\overline{T_{2i+1}}$ represents the upper bound on the final time of the events where low PND is allowed, and $\overline{T_{2i}}$ represents the upper bound on the final time of the events where high PND is forced. No linear trajectory or linear final constraints are used.

2) Nonlinear trajectory constraints

$$\frac{\underline{v}^2}{\bar{v}^2} \le W_v \frac{1}{T^2} \left[\left(\frac{\mathrm{d}n_{\mathrm{ac}}}{\mathrm{d}\tau} \right)^2 + \left(\frac{\mathrm{d}e_{\mathrm{ac}}}{\mathrm{d}\tau} \right)^2 \right] \le 1$$

$$-1 \le W_c \frac{(\mathrm{d}n_{\mathrm{ac}}/\mathrm{d}\tau)(\mathrm{d}^2 e_{\mathrm{ac}}/\mathrm{d}\tau^2) - (\mathrm{d}e_{\mathrm{ac}}/\mathrm{d}\tau)(\mathrm{d}^2 n_{\mathrm{ac}}/\mathrm{d}\tau^2)}{[(\mathrm{d}n_{\mathrm{ac}}/\mathrm{d}\tau)^2 + (\mathrm{d}e_{\mathrm{ac}}/\mathrm{d}\tau)^2]^{1.5}} \le 1$$

$$0 \le W_r[(n_{\mathrm{ac}} - n_{\mathrm{ref}})^2 + (e_{\mathrm{ac}} - e_{\mathrm{ref}})^2] \le 1$$

$$0 \le W_s[\max(s_i)] \le 1, \qquad i = 1, 2, \ldots, N_r$$

The first constraint puts a limit on the velocity of the UAV where $\underline{v}$ and $\bar{v}$ represent lower and upper limits of the velocity of the UAV, respectively. The next constraint is the curvature rate formulated to highlight the features of the NTG method for more realistic flight models. The third constraint helps UAV to follow an initial straight-line reference trajectory between waypoints within a specified radius. As noted in [1], the examples were sensitive to the initial coarse route, and it was difficult for the method to converge to a solution. This constraint aids in solution convergence by providing an acceptable bounded region for the solution. Finally, the last constraint is the signature constraint, which limits the maximum signature of the UAV among all present radar systems. During the low-PND region, $W_s = 1.0$ (maximum signature is allowed), whereas during the high-PND region, $W_s > 1.0$ (low signature is forced), i.e.,

if

$$T_{2i+1} \le t < T_{2i+2} \rightarrow W_s = 1.0$$

if

$$T_{2i} \le t < T_{2i+1} \rightarrow W_s > 1.0$$

Note that the constraint equations are scaled to aid in convergence of the nonlinear optimization codes. W_v, W_c, W_r, and W_s denote the corresponding weighting functions for each constraint.

3) Nonlinear final constraints

$$0 \le W_f[(n_{\mathrm{ac}}(1) - n_f]^2 + [e_{\mathrm{ac}}(1) - e_f)]^2 \le 1$$

$$\underline{\xi} \le W_d \cdot \tan^{-1}[(\mathrm{d}e_{\mathrm{ac}}/\mathrm{d}\tau)/(\mathrm{d}n_{\mathrm{ac}}/\mathrm{d}\tau)] \le \bar{\xi}$$

The UAV's arrival to the final destination waypoint (n_f, e_f) within a specified area is ensured by the first constraint. Note that this constraint, as all other constraint and cost functions, is also a function of the scaled time τ, and here $\tau = 1$ represents the final scaled time, i.e., if $t = T$ (T is the unknown real final time), then from $\tau = (t/T) \rightarrow \tau = 1$. A constraint on the final heading of the UAV to ensure that the aircraft is heading toward future waypoints is given by the next inequality. Note that ξ is the heading angle between waypoint $(k + 1)$ and waypoint (k). The lower and upper bounds on the heading angle are represented by $\underline{\xi}$ and $\bar{\xi}$, respectively.

The problem formulation given here is valid within two consecutive waypoints only. Therefore, the problem needs to be recast upon arrival at each waypoint by taking the current solution for this waypoint as the initial condition and the next waypoint as the target, until reaching the last waypoint, hence working as in receding horizon manner.

IV. Parameterization of Guidance

In trajectory generation problems, such as the one considered in this study, the state space of the vehicle is guided by at least one medium-specific effect, such as wind, current, etc. The designer needs to address these effects accurately in problem formulation regardless of the solution method.

In addition, specifically for the NTG solution, the user needs to specify 1) choice of outputs and their derivatives, 2) the cost and the constraints in terms of these outputs and their derivatives, and 3) the regularity of the variables, placement of the knot points, order and regularity of the B-splines, and collocation points for each output. Thus, NTG needs the derivatives of the probability of detection and signature functions with respect to the outputs. Numerically computing these derivatives directly from the data sets can easily create convergence problems.

An algebraic system to identify the generative analytical models of this given data is therefore necessary for the solver to evaluate the guidance at each point in the solution region. Because such evaluations are performed typically at every iteration along the solution process, this model is desired to be computationally cheap. Moreover, as in the case of NTG, some solvers may necessitate the approximation be not only continuous, but also continuously differentiable, possibly to several orders.

To satisfy the approximation need that arises, specifically in the case of using the NTG solver in trajectory generation problems, we suggest here feedforward ANNs to represent the guidance, i.e., to model the signature and probability of detection functions. We will compare the ANNs against B-spline functions, as well.

A. Brief Introduction to Feedforward Artificial Neural Network Approximator

The operation of a feedforward ANN is organized in functional levels called layers. It accepts input from its input nodes and each layer consists of simple computational units, each taking the outcome of the preceding layer and implementing a simple nonlinear algebraic input/output mapping

$$y = \phi\left(b + \sum_{i=1}^{n} w_i x_i\right) \tag{17}$$

where $x_1, \ldots, x_n$ are inputs, $\{w_i\}_{i=1}^{n} \subset \mathbb{R}$ are weight parameters, $b \in \mathbb{R}$ is the bias, and $\phi(\cdot)$ is the activation nonlinearity.

A popular choice for $\phi(\cdot)$ is a sigmoidal nonlinearity, such as $\tanh(\cdot)$. It is known that three consecutive layers of sigmoidal neurons suffice to approximate any algebraic function [23]. This is called the universal approximation property.

Given a finite set of input/output samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}$, $i = 1, \ldots, n$, the design of the ANN approximator can be achieved by the backpropagation training rule, which is simply a gradient-descent algorithm minimizing the training error

$$E(\mathcal{W}) = \sum_{i=1}^{n} \|\mathbf{y}_i - \Gamma(\mathcal{W}, \mathbf{x}_i)\|_2^2 \tag{18}$$

where $\mathcal{W}$ is the set of all parameters of the network and $\Gamma(\cdot)$ is the network transfer function, typically in the form of nested sigmoids.

In the demonstrations presented in the following, we consider a three-layer network where the first two (hidden) layers employ sigmoidal nonlinearity and the output layer consists of a single linear unit, simply taking the weighted sum of its inputs, as illustrated in Fig. 4, by omitting network parameters.

B. Approximation of Signature and Probability of Detection Functions Using Artificial Neural Networks

We considered a feedforward ANN architecture (cf. Fig. 4) for each of the two variables guiding the state space [8]. For probability of detection function $\mathrm{PD}(s, r)$, a network employing 5 units in the first and 3 units in its second sigmoidal layer was trained using the conventional backpropagation algorithm. The network was able to fit to the data in Table 2 to an error on the order of 10^{-7} after training. As a result, the approximating function for Table 2 was determined by a function of the form

$$\widehat{\mathrm{PD}}(s, r) = \mathbf{W}_3 \Theta\left[\mathbf{W}_2 \Theta\left(\mathbf{W}_1 \begin{bmatrix} s \\ r \end{bmatrix} + \mathbf{b}_1\right) + \mathbf{b}_2\right] + \mathbf{b}_3 \tag{19}$$

where $\{\mathbf{W}_i, \mathbf{b}_i\}$, $i = 1, 2, 3$, are the weights and biases of the three layers in matrix form, and $\Theta(\cdot)$ is the diagonal $\tanh(\cdot)$ transform. It should be noted that, due to the considered network architecture, the resulting function is continuously differentiable, without requiring any additional constraints on training.

A similar network architecture was considered for the signature function $s(\varepsilon, \alpha)$ (Table 1) with 7 and 5 units in its first and second sigmoidal layers, respectively. The training error was on the order of

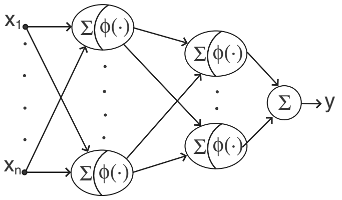
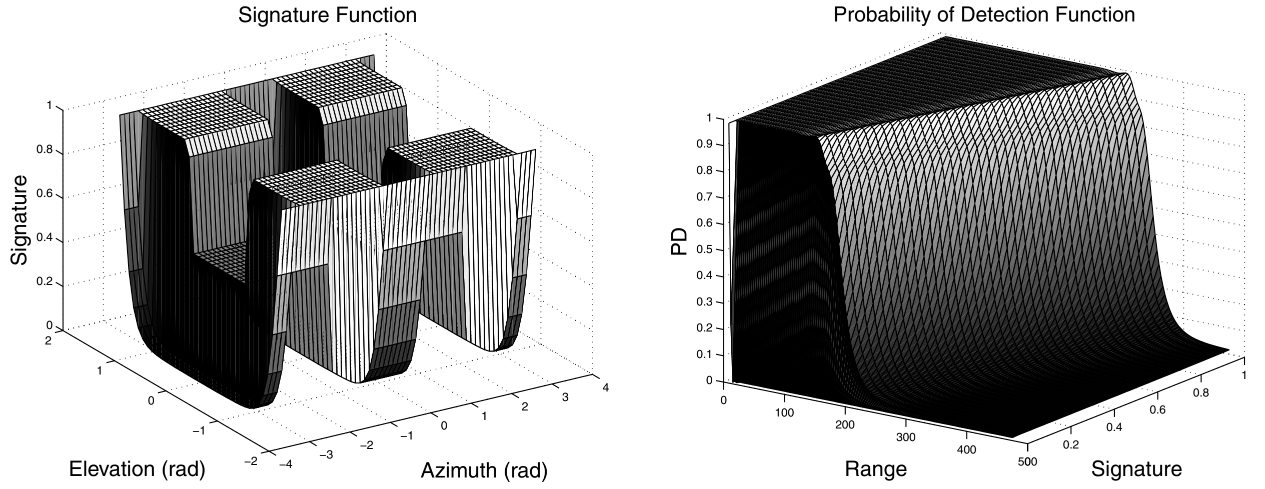
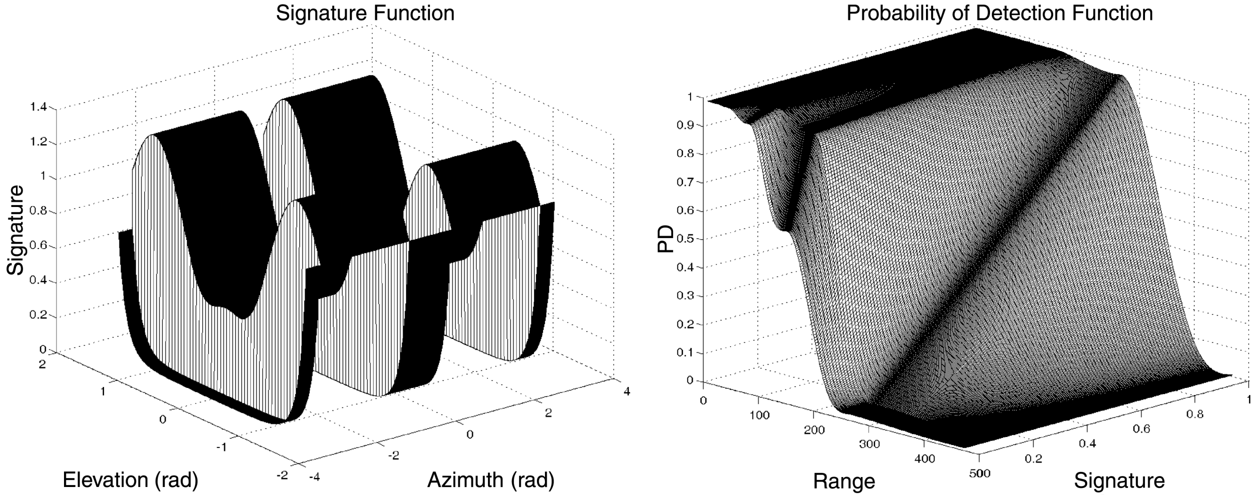


Fig. 4 Feedforward ANN architecture used to identify guidance in the optimal control problem.



a) Signature model **b) Probability of detection model**

Fig. 5 Approximations of signature and probability of detection functions obtained by three-layer sigmoidal ANNs.



a) Signature model **b) Probability of detection model**

Fig. 6 Approximations of signature and probability of detection functions obtained by B-spline tensor product functions.

10^{-6}. The resulting form was the same as the one in Eq. (19), with the input vector $[s \quad r]^T$ replaced by $[\varepsilon \quad \alpha]^T$.

It is important to notice that when azimuth angle changes between ∓ 30 deg (∓ 0.524 rad) and ∓ 31 deg (∓ 0.541 rad), there is a big change in the magnitude of the signature function of the UAV, as it is shown in Table 1. The same phenomena occurs when the azimuth angle changes between ∓ 159 deg (∓ 2.775 rad) and ∓ 160 deg (∓ 2.793 rad). This introduces several local minima and sharp changes in the gradients. Thus, the underlying optimization problem is, in fact, quite difficult. Plots of the resulting two functions are given in Fig. 5. Note the sharp changes in signature for azimuths around the aforementioned regions.

Having approximated the PD and s tables by the functions depicted in Fig. 5, which are differentiable with respect to each of their two variables, the NTG framework can now be used to perform the optimization given in the preceding subsection.

C. Approximation of Signature and Probability of Detection Functions Using B-Splines

For comparison with the ANN approximation of guidance for signature and probability of detection functions, we employed the tensor product B-spline functions [9].[††]

Analytical models to the signature s and the probability of detection tables, given in Tables 1 and 2, can be found efficiently using tensor product B-spline functions:

$$s = f_s(\alpha, \varepsilon) = \sum_{i=1}^{g} \sum_{j=1}^{h} B_{i,k_\alpha}(\alpha) B_{j,k_\varepsilon}(\varepsilon) a_{ij}$$

$$\mathrm{PD} = f_{pd}(s, r) = \sum_{i=1}^{p} \sum_{j=1}^{q} B_{i,k_s}(s) B_{j,k_r}(r) b_{ij}$$

Figure 6 shows the result of the fit function models by B-spline tensor product functions vs the actual data points indicated by "o". The order of the polynomials in the models used are $k_\alpha = 4$, $k_\varepsilon = 2$, $k_s = 2$, and $k_r = 4$, and the number of coefficients are $g = 9$, $h = 11$, $p = 7$, and $q = 30$. An accuracy on the order of 10^{-7} can be achieved in the B-spline approximation, as in the case of ANNs, at the cost of an increased risk of overfitting. This can be observed by comparing Figs. 5 and 6.

Implementing the models in the NTG formulations requires care. The models can be implemented by writing a subroutine that evaluates the value of the models and their derivatives with respect to a given location of the aircraft. Interested readers are referred to de Boor [24]. Implementation of derivatives of the models, for example for signature function, should be done using the chain rule

$$\alpha = f_\alpha[\psi(\dot{n}_{\mathrm{ac}}, \dot{e}_{\mathrm{ac}}), n_{\mathrm{ac}}, e_{\mathrm{ac}}] \qquad \varepsilon = f_\varepsilon[\psi(\dot{n}_{\mathrm{ac}}, \dot{e}_{\mathrm{ac}}), n_{\mathrm{ac}}, e_{\mathrm{ac}}]$$

[††] Note that the derivatives of the guidance models are required in the NTG formulation. These are straightforward to calculate in B-spline as well as in ANN formulations.

$$
\begin{aligned}
\frac{\mathrm{d}s}{\mathrm{d}n_{\mathrm{ac}}} &= \frac{\mathrm{d}s}{\mathrm{d}\alpha}\cdot\frac{\mathrm{d}\alpha}{\mathrm{d}n_{\mathrm{ac}}} + \frac{\mathrm{d}s}{\mathrm{d}\varepsilon}\cdot\frac{\mathrm{d}\varepsilon}{\mathrm{d}n_{\mathrm{ac}}}\\
\vdots\quad & \quad\vdots \qquad\qquad \vdots\\
\frac{\mathrm{d}s}{\mathrm{d}\dot{e}_{\mathrm{ac}}} &= \frac{\mathrm{d}s}{\mathrm{d}\alpha}\cdot\frac{\mathrm{d}\alpha}{\mathrm{d}\dot{e}_{\mathrm{ac}}} + \frac{\mathrm{d}s}{\mathrm{d}\varepsilon}\cdot\frac{\mathrm{d}\varepsilon}{\mathrm{d}\dot{e}_{\mathrm{ac}}}
\end{aligned}
$$

Derivatives of the PD, and hence PND, function can be implemented similarly.

V. Numerical Examples

To investigate the proposed approach, several examples are presented in this section. It is assumed that pitch and bank angles are zero and altitude is fixed at 12 km in this study. Therefore, the considered domain is represented as 2D Cartesian coordinates in kilometers. There are six temporal events used between each waypoint in all examples: three for high-PND regions, and three for low-PND regions. It is important to mention that the radar regions are *not* modeled as obstacles in any examples. The only assumption is that the locations, types, and engagement zones of radars are known, so that the elevation and azimuth angles can be calculated and appropriate signature and PD tables can be used. Outside of the engagement zones of the radars, which are shown with circular regions in the examples with dashed lines, the signature and probability of detection are assumed to be at their minimum values. In this section, *B* and *T* in the figures represent the base and target locations, respectively.

The first example was set on a domain containing two radars located at (0, 0 km) and (100, 100 km), and two waypoints at (−50, −50 km) (the base location of UAV) and (150, 150 km) (the target), as illusturated in Fig. 7. The ground track of the low-observable trajectory, generated by NTG in cooperation with the proposed ANN approximator, is shown with a dotted solid line. The dashed boundaries indicate the range of radars, i.e., the regions with detection probabilities PD > 0.01. However, these regions were not modeled as obstacles in this problem and these regions are included for illustration purpose only, as mentioned before. The resulting *low-observable* trajectory successfully avoids the radar detection systems. A straight-line trajectory would go directly through the centers of the radar detection systems in this simulation. On the other hand, the low-observable trajectory skirts radar engagement zones. Note that this trajectory satisfies the flyability constraints given in the previous section, including radius of curvature.

Figure 8 shows the result of another example with two radars located at (−60, −100 km) and (−60, −15 km), and two waypoints at (−100, −40 km) and (−20, −0 km). Note that this is the shortest path avoiding the radar regions because the optimization also tries to minimize the total flight time.

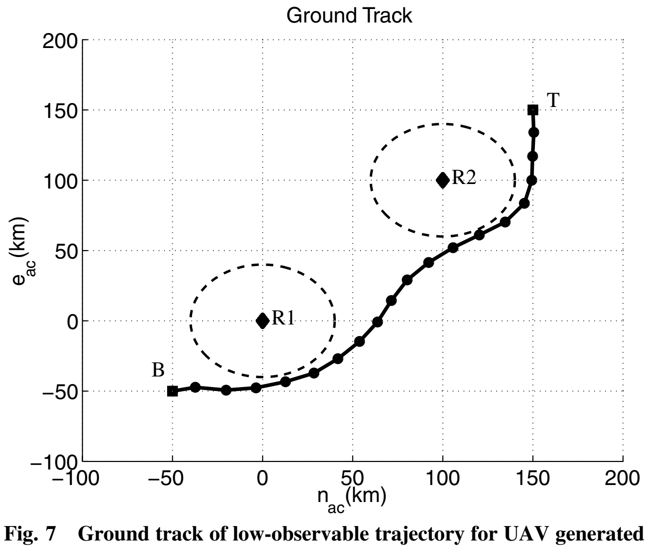


Fig. 7 Ground track of low-observable trajectory for UAV generated by NTG with ANN approximation in the first experiment with two radars and two waypoints.

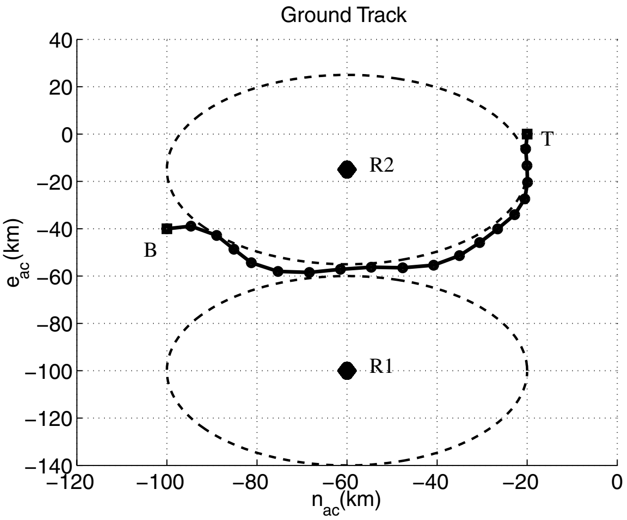


Fig. 8 Ground track of low-observable trajectory for UAV generated by NTG with ANN approximation in the second experiment with two radars and two waypoints.

The last example is to find a low-observable optimal trajectory from a base location (−100, −100 km) to three different target sites located within radar regions at (0, 30 km), (200, 20 km), and (−200, −130 km). Therefore, an optimal trajectory intersects such regions, minimizing the trajectory cost function defined in the previous section. Intermediary course waypoints are located at (−46, −30 km), (100, 25 km), and (200, −40 km). These initial course waypoints construct initial straight-line trajectories between the base point and first target, first and second targets, and finally the second and last targets, respectively. Radars are located at (0, 0 km), (200, 50 km), and (200, −100 km). In this experiment, the trajectory generation problem was programmed into NTG in six consecutive subproblems, whereas the guidance due to three radars was identified by the proposed ANN approach as done in the previous experiments. Figure 9 clearly illustrates that the UAV finds optimum entrance approach to the radar zones toward the targets to optimize its observability against radar systems and the total flight time. Note that, in the optimum entrance approach, the UAV flies with low azimuths. To reach targets inside the radar engagement zones, these zones are entered and exited with optimized azimuths, "nose-in" for entry and "tail-in" for exit. In this figure, *B*, *T*1, *T*2, *T*3, *W*1, *W*2, and *W*3 represent the base, three different target locations, and other initial waypoints, respectively.

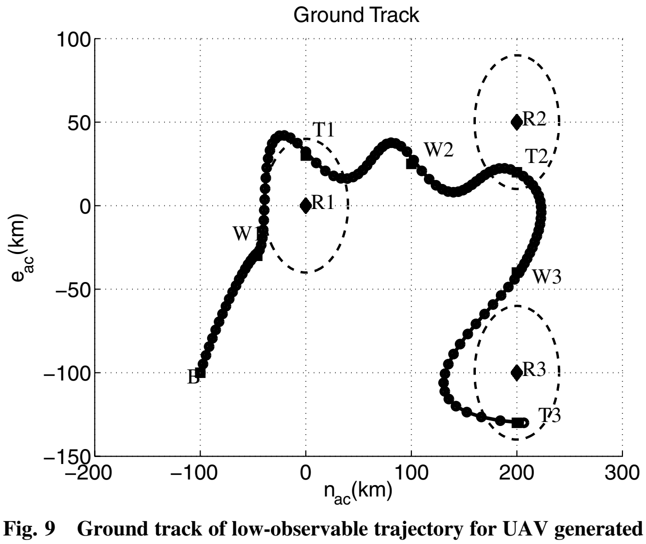


Fig. 9 Ground track of low-observable trajectory for UAV generated by NTG and ANN approximator in the third experiment with three radars and seven waypoints.

The trajectories illustrated in Figs. 7–9 are near-optimal, locally minimizing the flight length and observability by the radars. On the other hand, Figs. 10–12 show the results of the same experiments, whereas the guidance due to radar(s) was identified by the tensor product B-spline functions as presented in Sec. IV.C. The effect of B-spline overfitting is clear in these figures.

An obvious difference between these trajectories is their smoothness. The underlying reason of this difference is the simplicity of feedforward ANN architecture and design relative to that of B-splines. In particular, the ANN architecture introduces fewer parameters for the same approximation task that are all continuously differentiable throughout the parameter space. ANN employes 36 parameters for the PD and 66 for the *s* models. On the other hand, there are 210 and 99 B-spline parameters for the PD and *s* models, respectively. Moreover, its training procedure performs an unconstrained minimization. On the other hand, B-spline design imposes continuity constraints on the piecewise polynomials. It also requires potentially many trial-and-error steps before the parameter adjustments phase, because continuity and flatness conditions impose additional constraints on the analytical form of the piecewise polynomial. Because of these differences, design of the feedforward ANN approximator is computationally more efficient than the B-spline design.

Finally, note that it is possible to trim the network architecture and minimize the complexity of the ANN approximator even further [25]. Though not demonstrated in this study, such an extension of the guidance approximation phase would avoid altogether trial-and-error search of an appropriate network structure.

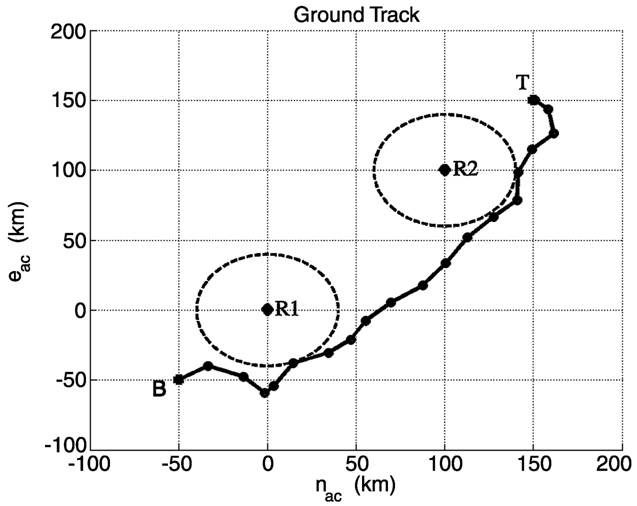


Fig. 10 Ground track generated by NTG and B-spline approximator in first experiment.

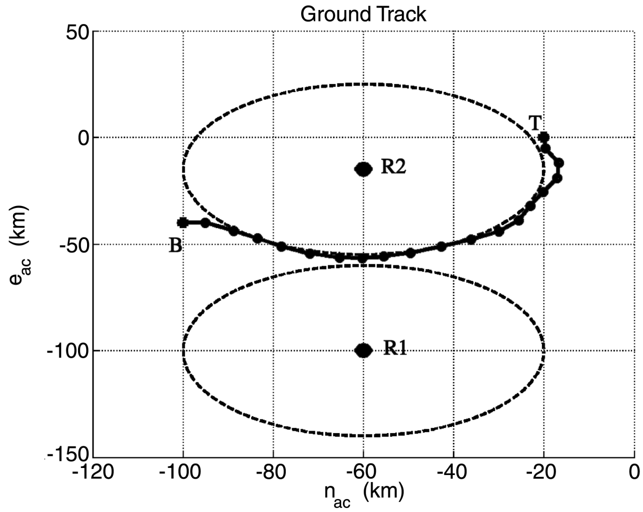


Fig. 11 Ground track generated by NTG and B-spline approximator in second experiment.

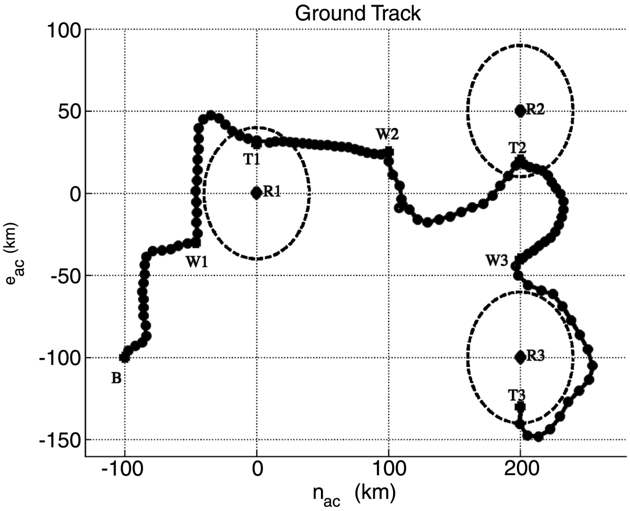


Fig. 12 Ground track generated by NTG and B-spline approximator in third experiment.

VI. Conclusions

In this paper, we proposed a method of finding real-time, low-observable nonlinear trajectories for unmanned aerial vehicles to optimize a weighted function of total mission time, speed, and probability of detection, so that risk of the aircraft is minimized against radar detection systems.

We suggested the use of feedforward ANN approximation in identifying the guidance on a state space as a differentiable function to be evaluated by a dynamic optimization process. In comparison, the same guidance (analytical models of the signature and probability of detection tables) was calculated using the tensor product B-spline functions. Then, the trajectory generation problem and its implementation in the NTG were explained in detail. Illustrative numerical examples that indicate the ability of our proposed method to find real-time, low-observable nonlinear optimal trajectories from a start point to a final destination point, between radar detection systems, as well as inside radar regions, were provided. The resulting trajectories obtained by using the proposed feedforward ANN approximator are smoother and shorter than the ones obtained by B-spline approximation.

We noticed that, as the complexity of the problem increases by using the probability of detection table with the signature table, this leads to more complicated convergence issues. Our future work will focus on convergence issues and on extending the proposed method to three dimensions.

Acknowledgments

This research is based upon work supported by the Defense Advanced Research Projects Agency Advanced Research Projects, Information Exploitation Office, and the U.S. Air Force Research Laboratory under Contract No. F33615-01-C-3149. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency or the U.S. Air Force. The authors gratefully acknowledge the helpful comments of the reviewers as well as the Defense Advanced Research Projects Agency and U.S. Air Force Research Laboratory management.

References

[1] Misovec, K., Inanc, T., Wohletz, J., and Murray, R. M., "Low-Observable Nonlinear Trajectory Generation for Unmanned Air Vehicles," *Proceedings of the 42nd IEEE Conference on Decision and*

Control, Vol. 3, Inst. of Electrical and Electronics Engineers, New York, Dec. 2003, pp. 3103–3110.

[2] Norsell, M., "Radar Cross Section Constraints in Flight Path Optimization," *Journal of Aircraft*, Vol. 40, No. 2, 2003, pp. 412–415.

[3] Kim, J., and Hespanha, J. P., "Discrete Approximation to Continuous Shortest-Path: Application to Minimum-Risk Path Planning for Groups of UAVs," *Proceedings of the 42nd IEEE Conference on Decision and Control*, Vol. 2, Inst. of Electrical and Electronics Engineers, New York, Dec. 2003, pp. 1734–1740.

[4] Novy, M. C., Jacques, D. R., and Pachter, M., "Air Vehicle Optimal Trajectories Between Two Radars," *Proceedings of the American Control Conference*, Vol. 1, American Automatic Control Council, Evanston, IL, May 2002, pp. 785–790.

[5] McFarland, M., Acjeru, R., and Taylor, B., "Motion Planning for Reduced Observability of Autonomous Aerial Vehicles," *Proceedings of the IEEE International Conference on Control Applications* Vol. 1, Inst. of Electrical and Electronics Engineers, New York, 1999, pp. 231–235.

[6] Pachter, M., and Hebert, F., "Optimal Aircraft Trajectories for Radar Exposure Minimization," *Proceedings of the American Control Conference*, Vol. 3, American Automatic Control Council, Evanston, IL, 2001, pp. 2365–2369.

[7] Corman, D., and Knutti, J., "MICA OEP User Guide 1.1," Boeing Co. Tech Rept., 2003.

[8] Muezzinoglu, M. K. and Inanc, T., "Trajectory Generation in Guided Spaces Using Artificial Neural Networks and NTG Algorithm," *Proceedings of the American Control Conference*, American Automatic Control Council, Evanston, IL, 2006, pp. 5776–5781.

[9] Inanc, T., Misovec, K., and Murray, R. M., "Nonlinear Trajectory Generation for Unmanned Air Vehicles with Multiple Radars," *Proceedings of the 43th IEEE Conference on Decision and Control*, Vol. 4, Inst. of Electrical and Electronics Engineers, New York, Dec. 2004, pp. 3817–3822.

[10] Hornik, K., Stinchcombe, M., and White, H., "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, No. 5, 1989, pp. 359–366.
doi:10.1016/0893-6080(89)90020-8

[11] Werbos, P. J., "Backpropagation Through Time: What It Does and How to Do It," *Proceedings of the IEEE*, Vol. 78, No. 10, Oct. 1990, pp. 1550–1560.
doi:10.1109/5.58337

[12] Muezzinoglu, M. K., and Zurada, J. M., "Projection-Based Gradient Descent Training of Radial Basis Function Networks," *Proceedings of IEEE International Joint Conference on Neural Networks*, Vol. 2, Inst. of Electrical and Electronics Engineers, New York, July 2004, pp. 1297–1302.

[13] Milam, M., "Missile Interception Research Report," California Inst. of Technology Internal Rept., 2002.

[14] Lian, F.-L., and Murray, R., "Cooperative Task Planning of Multi-Robot Systems with Temporal Constraints," *Proceedings of the International Conference on Robotics and Automation, IEEE International Conference on Robotics and Automation*, Vol. 2, Inst. of Electrical and Electronics Engineers, New York, 2003, pp. 2504–2509.

[15] Milam, M. B., "Real-Time Optimal Trajectory Generation for Constrained Dynamical Systems," Ph.D. Thesis, California Inst. of Technology, Pasadena, CA, 2003.

[16] Gill, P. E., Murray, W., Saunders, M., and Wright, M., *NPSOL: Nonlinear Programming Software*, Stanford Business Software, Inc., Mountain View, CA, 30 July 1998.

[17] Murray, R. M., Rathinam, M., and Sluis, W., "Differential Flatness of Mechanical Control Systems: A Catalog of Prototype Systems," *ASME International Mechanical Engineering Congress and Expo*, Nov. 1995.

[18] Bryson, A. E. and Ho, Y. C., *Applied Optimal Control: Optimization, Estimation and Control*, Taylor and Francis, Washington, D.C., 1975.

[19] Fliess, M., Levine, J., Martin, P., and Rouchon, P., "Flatness and Defect of Nonlinear Systems: Introductory Theory and Examples," *International Journal of Control*, Vol. 61, No. 6, 1995, pp. 1327–1360.
doi:10.1080/00207179508921959

[20] Milam, M., Mushambi, K., and Murray, R., "New Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems," *Proceedings of the 39th IEEE Conference on Decision and Control*, Vol. 1, Inst. of Electrical and Electronics Engineers, New York, Dec. 2000, pp. 845–851.

[21] Milam, M., Franz, R., and Murray, R., "Real-Time Constrained Trajectory Generation Applied to a Flight Control Experiment," *Proceedings of IFAC*, International Federation of Automatic Control, 2002.

[22] Lian, F.-L., and Murray, R., "Real-Time Trajectory Generation for the Cooperative Path Planning of Multi-Vehicle Systems," *Proceedings of the IEEE Conference on Decision and Control*, Vol. 4, Inst. of Electrical and Electronics Engineers, New York, 2002, pp. 3766–3769.

[23] Haykin, S., *Neural Networks: A Comprehensive Foundation*, 2nd ed., Prentice–Hall, Upper Saddle River, NJ, 1999.

[24] de Boor, C., *Practical Guide to Splines*, Springer-Verlag, New York, 2001.

[25] Zurada, J. M., *Introduction to Artificial Neural Systems*, West Publishing, St. Paul, MN, 1992.