Submitted, 2004 Conference on Decision and Control
http://www.cds.caltech.edu/~murray/papers/2004j_imm04-cdc.html

Nonlinear Trajectory Generation for Unmanned Air Vehicles with Multiple Radars

Tamer Inanc, Kathy Misovec and Richard M. Murray

***Abstract*— The problem of finding a real time optimal trajectory to minimize the probability of detection (to maximize the probability of not–being–detected, pnd, function) of unmanned air vehicles by opponent radar detection systems is investigated. This paper extends our preliminary results on low observable trajectory generation in three ways. First, trajectory planning in the presence of detection by multiple radar systems, rather than single radar systems, is considered. Second, an overall probability of detection function is developed for the multiple radar case. In previous work, both probability of detection by a single radar and signature were developed in the theory section, but the examples used only signature constraints. In this work, the use of the overall probability of detection function is used, both because it aids in the extension to multiple radar systems and because it is a more direct measure of the desirable optimization criteria. The third extension is the use of updated signature and probability of detection models. The new models have a greater number of sharp gradients than the previous models, with low detectability regions for both a cone shaped areas centered around the nose as in the previous paper, as well as a cone-shaped area centered around rear of the air vehicle. The Nonlinear Trajectory Generation method (NTG), developed at Caltech, is used and motivated by the ability to provide real time solutions for constrained nonlinear optimization problems. Numerical simulations of multiple radar scenarios illustrate UAV trajectories optimized for both detectability and time.**

I. INTRODUCTION

One of the main threats for Unmanned Air Vehicles (UAVs) is radar detection systems. Designing optimal trajectories for UAV's which reduces their detectability against radar systems has been attracting much attention in the last few years. In [18], we find low–observable trajectories for a UAV in the presence of a single radar without modeling the radar as an obstacle region. Martin Norsell et al. [19] uses similar approach to our method. He uses radar cross section (RCS) constraints in flight path optimizations. The optimization problem is separated into two stages and the radar detection constraint is invoked during the first stage only, on the other hand, the minimum time constraint is invoked during the second stage. In [17], Kim and Hespanha addressed the weighted anisotropic shortest–path problem on a continuous domain and applied it to the computation of paths for UAVs to minimize the risk of being destroyed by ground defenses. In [20], the feasibility of using geometric, deterministic solutions to optimal trajectories between two radars is explored. The optimal paths are compared with the direct path and trajectories found by Voronoi path planning approach and it is limited to exploring the constant–velocity, single vehicle, constant RCS scenario. Work by McFarland et al. [15] uses motion planning techniques using potential field theory for unmanned air vehicle (UAV) path planning in the presence of detection systems. This is a technique originally used in robot motion planning. An analytical solution yielding the trajectory that minimizes the radar energy reflected from the target is derived using the Calculus of Variations in [21].

In this work, we investigate the problem of finding a real time optimal trajectory to minimize the probability of detection function (to maximize the probability of not–being–detected, pnd, function) of unmanned air vehicles by opponent radar detection systems. The aim is to find a real time nonlinear optimal trajectory maximizing the pnd function of UAV's against hostile radar detection systems as well as minimizing the total flight time between a given base station and a final destination. Thus, maximizing the survivability of the aircraft. This work extends our preliminary results on low observable trajectory generation in three ways. First, trajectory planning in the presence of detection by multiple radar systems, rather than single radar systems, is considered. Second, an overall probability of detection function is developed for the multiple radar case. In previous work, both probability of detection by a single radar and signature were developed in the theory section, but the examples used only signature constraints. In this work, the use of the overall probability of detection function is used, both because it aids in the extension to multiple radar systems and because it is a more direct measure of the desirable optimization criteria. The third extension is the use of updated signature and probability of detection models. The new models have a greater number of sharp gradients than the previous models, with low detectability regions for both a cone shaped areas centered around the nose as in the previous paper, as well as a cone-shaped area centered around rear of the air vehicle.

The first, we find analytical models for the probability of not–being–detected (pnd) function using tensor product B–Splines. Then, we define two different event types where in the first one maximum pnd function must be satisfied, and in the second one *limited* time high–observability (minimum pnd) periods to the radars might be allowed.

T. Inanc is a postdoctoral scholar at California Institute of Technology, 1200 E. California Blvd. Pasadena, CA 91125, (contact author) tinanc@cds.caltech.edu

K. Misovec is with ALPHATECH Inc. 6 New England Executive Park Burlington, MA 01803

R. M. Murray is with Faculty of California Institute of Technology, 1200 E. California Blvd. Pasadena, CA 91125, murray@cds.caltech.edu

Hence, by interspersing maximum pnd times where the UAV's observability to the radar systems is minimized with the short periods of high observability regions it might be possible to find optimal trajectories among the radar regions and as well as in the radar regions. This is achieved by driving the radar systems into a condition called *lock–loss* in which a radar system aborts its plans after a specified time of no detection. The Nonlinear Trajectory Generation (NTG) algorithm developed at Caltech by Mark Milam et al. [1] is, then, used to generate real–time trajectories achieving these goals with other constraints. The NTG algorithm first finds flat outputs using differential flatness property to reduce the problem dimension, and then parameterizes these outputs using B-Spline curves. The coefficients of the B-Spline curves are further solved by the sequential quadratic programming to satisfy the optimization objectives and constraints.

The paper is organized as follows. The second section gives the necessary background on aircraft and detection models, NTG and tensor product B–Splines. Next section introduces formulation of the trajectory generation problem which maximizes the pnd function while also minimizes the total mission time. This section also gives a brief information on how to use tabular data modeled by B–Splines in the NTG formulation. Numerical simulations with multiple radars are provided in the fourth section. Finally, the last section gives the conclusions and future work.

II. BACKGROUND

In this section, brief introductions to B–Spline functions, Nonlinear Trajectory Generation software package, aircraft and detection models are given.

A. Tensor Product B–Spline Functions

Spline functions have been extensively used as an approximation tool in areas such as curve fitting, computer graphics, image and signal processing, motion planning and trajectory generation for mechanical systems [9], [10]. For a detailed information, the reader is directed to the book by Carl de Boor [11]. Neuman and Sen in [12] were one of the earliest researchers to apply cubic splines to solve optimal control problems. They used cubic spline collocation on a uniform mesh to solve several optimal control problems with quadratic objective functions. Ever since then, splines have been used extensively to formulate and solve optimal control problems numerically.

Although polynomials are favorable to approximate functions due to their ease of evaluations using simple arithmetic operations, they have two major problems: (1) if a function to approximate is in a large interval then the approximating polynomial order can be quite large, (2) polynomials have the global dependence on local parameters, i.e., if a fit is poor in a small area then it will be poor in whole curve [11]. Solution is to use *piecewise* polynomials.

Piecewise polynomials offer a local support, each segment of the piecewise polynomial approximates the trajectory and satisfies the constraints locally, and it can be achieved with relatively low order polynomials. Therefore, they are more suitable for optimal control problems. It is also possible to patch the polynomial segments in such a way so that the resulting trajectory has several continuous derivatives at all points. Any smooth piecewise polynomial is called a *Spline*.

Splines are most conveniently constructed when expressed as B-Splines. B-Spline curves are generalizations of $B\acute{e}zier$ curves. It is constructed by joining several $B\acute{e}zier$ curves with a prescribed level of continuity between them. The points at which the curves are joined are called *break points*. The nondecreasing list of real numbers containing the breakpoints is called a *knot vector*. The *smoothness* s_i of a breakpoint provides the level of continuity at a breakpoint. A breakpoint is $(s_i - 1)$ times continuously differential.

A trajectory $z(x)$, with prescribed smoothness s and k^{th} order of the piecewise polynomials can be written as

$$z(x) = \textstyle\sum_{i=1}^{p} B_{i,k} C_i$$

where the number of C_i coefficients is given by $p = l(k-s)+s$, which can be used to customize the trajectory, l represents the number of intervals. The functions $B_{i,k}(x)$ are B-Spline basis functions defined by

$$B_{i,1}(x) \quad = \quad \begin{cases} 1 & \textit{if} \quad t_i \le x < t_{i+1} \\ 0 & otherwise \end{cases} \tag{1}$$

$$\text{if} \qquad t_i = t_{i+1} \quad B_{i,1} = 0$$

Higher order terms can be found using *Cox-de Boor* recursion formula:

$$B_{i,k}(x) = \tfrac{x-t_i}{t_{i+k-1}-t_i} B_{i,k-1}(x) + \tfrac{t_{i+k}-x}{t_{i+k}-t_{i+1}} B_{i+1,k-1}(x) \ , \ k > 1. \tag{2}$$

The derivatives of the B–Spline functions, which is necessary to use in the NTG formulation, are also easy to calculate. For example, the first derivative of a k^{th} order B–Spline function is a $(k-1)^{th}$ order B–Spline function and is given by

$$\text{If} \quad \sum_i B_{i,k} C_i \ = \sum_{i=r-k+1}^{s-1} B_{i,k} C_i \quad \textit{on} \ [t_r \cdots t_s]$$

$$D\left(\sum_i B_{i,k} C_i\right) = \sum_{i=r-k+2}^{s-1} (k-1) \frac{C_i - C_{i-1}}{t_{i+k-1} - t_i} B_{i,k-1} \text{ on } [t_r \cdots t_s]$$

where D represents the first order derivative. Finally, 2D tensor product B–Spline functions which we use to find

analytical models for radar signature and probability of detection tables in the next sections are given by

$$z(x,y) = \sum_{i=1}^{P_x} \sum_{j=1}^{P_y} B_{i,k_x}(x) B_{j,k_y}(y) C_{i,j}$$

Derivatives of the tensor product B–Spline functions can be calculated as in 1D case.

B. Nonlinear Trajectory Generation Algorithm (NTG)

The software package called Nonlinear Trajectory Generation (NTG) designed at Caltech by Mark B. Milam et al. [1] to solve optimal control problems in real-time is based on three components; (1) Parameterization to map system dynamics to lower dimensional space using differential flatness property (it will be easier and computationally more efficient to solve a lower dimensional problem), (2) B-Splines as basis functions for parameterization (B-Splines are used as basis functions for their flexibility and calculating their derivatives), (3) Sequential quadratic programming to solve for B-Spline coefficients to minimize the desired cost function subject to the linear and nonlinear constraints (the sequential quadratic programming package NPSOL by Gill et al. [8] is used as the nonlinear programming solver in NTG). The user specifies a problem to NTG by stating the problem in terms of some choice of outputs and its 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 collocation points for each output.

The main advantages of the NTG over other techniques is that NTG produces approximate but very fast, optimal trajectories which makes it very useful for real-time applications. In addition to that, linear as well as nonlinear constraints (temporal and spatial) and cost functions can be included in the problem formulation of NTG. Specifically, these constraints and cost functions are structured in NTG formulation as: (1) Linear Initial Constraints, (2) Linear Trajectory Constraints, (3) Linear Final Constraints, (4) Nonlinear Initial Constraints, (5) Nonlinear Trajectory Constraints, (6) Nonlinear Final Constraints, (7) Initial Cost Function, (8) Trajectory Cost Function, (9) Final Cost Function. In this paper, the use of NTG is also motivated by recent extensions in NTG's ability to deal with temporal constraints [4] as well as tabular data (analytically modeled by B–Splines) constraints [18]. Other applications of NTG to several control problems can be found in [2], [3] and [6].

C. Aircraft Model

The aircraft and the detection models, as shown in Fig. (1), are the two main components based on the Open Experimental Platform (OEP) [7]. Although the OEP model allows for more complexity, we use here simplified models. We also assume that (1) the UAV maintains a constant altitude and the pitch angle is zero, (2) the bank angle

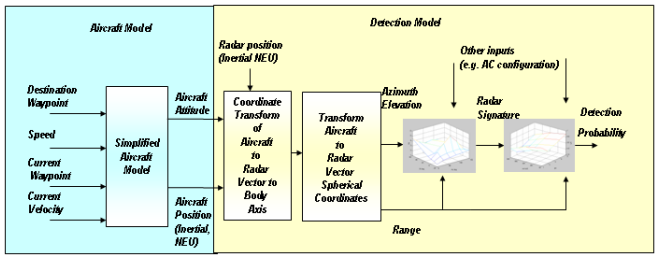

Fig. 1. The Aircraft and Detection Models.

is zero, (3) radars are mono–static. The first assumption implies that the azimuth effects are more dominant than the elevation effects. This assumption is more realistic at stand-off ranges. Elevation has a more pronounced effect as the aircraft gets closer in distance to the radar. The second assumption derives from assuming that the turn times are small compared to the time between waypoints. This assumption becomes less valid as the waypoint routes become finer. Mono–static radars consist of a collocated transmitter and receiver, and each station uses a different frequency so that there is no interaction between observers.

Current position, velocity, heading and the destination position are the inputs to the aircraft model, while the outputs are the aircraft position and attitude in inertial coordinates $(North, East, Up)$. The point mass model equations below represent the state equations for the vehicle traveling from waypoint i to the next waypoint $i+1$.

$$\begin{aligned} \dot{n}_{ac} &= U_i cos(\psi_i) \\ \dot{e}_{ac} &= U_i sin(\psi_i) \\ \dot{u}_{ac} &= 0 \qquad (3) \\ \psi_i &= tan^{-1} \frac{\dot{e}_{ac}}{\dot{n}_{ac}} \qquad (4) \end{aligned}$$

where n_{ac}, e_{ac}, and u_{ac} are the aircraft positions along the north, east and up axes, respectively. U_i is the speed and ψ_i 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_{ac}, e_{ac}, u_{ac}) to the radar (n_R, e_R, u_R) is given in body axes as;

$$\begin{bmatrix} x_{Rac} \\ y_{Rac} \\ z_{Rac} \end{bmatrix} = \begin{bmatrix} cos(\psi) & sin(\psi) & 0 \\ -sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (n_R - n_{ac}) \\ (e_R - e_{ac}) \\ (u_R - u_{ac}) \end{bmatrix} \qquad (5)$$

D. Detection Model

As shown in Fig. (1), the inputs to the detection model are the azimuth, elevation and slant range (az, el, R_s). To obtain these values the vector in (5) is transformed to spherical coordinates as follows

$$az = tan^{-1} \left(\frac{-(n_R - n_{ac})\dot{e}_{ac} + (e_R - e_{ac})\dot{n}_{ac}}{(n_R - n_{ac})\dot{n}_{ac} + (e_R - e_{ac})\dot{e}_{ac}} \right) \qquad (6)$$

$$el = tan^{-1}\left(\frac{u_R - u_{ac}}{\sqrt{(n_R - nac)^2 + (e_R - e_{ac})^2}}\right) \quad (7)$$

$$R_s = \sqrt{(n_R - nac)^2 + (e_R - e_{ac})^2 + (u_R - U_{ac})^2} \quad (8)$$

For the next part of the detection model, radar signature and probability of detection tables are used. The tables depend on the type of aircraft, the configuration of the aircraft, as well as the type of radar. In this paper, we use tables based on the large UAV with normal configuration and medium Surface Air Missile (SAM) radar model parameters. Details of the aircraft and detection models can be found in [7] and [18]. The first table computes the radar signature given the azimuth and elevation angles in radians. The signature is a unitless, intermediary variable that is related to radar cross section. The second table computes the probability of detection function given the radar signature and range values. In the next section, tensor product B–Spline functions are used to find analytical models to these tables to compute probability of not–being–detected, pnd, function.

III. TRAJECTORY GENERATION

This section introduces formulation of the trajectory generation problem which maximizes the probability of not–being–detected (pnd) function while also minimizes the total mission time. This section also gives a brief information on how to use tabular data modeled by B–Splines in the NTG formulation.

A. Problem Formulation

The problem is to find a real–time nonlinear trajectory of a UAV which maximizes the survivability of the aircraft against radar detection systems in minimum flight time given that radar locations and types, an aircraft type and configuration, a start location and initial heading angle of the aircraft and a destination location are known. First, let's define two types of events;

- Probability of not–being–detected function (pnd) *must be* high between $T_{2i-1} \leq t \leq T_{2i}$
- Probability of not–being–detected function (pnd) *could be* low between $T_{2i} \leq t \leq T_{2i+1}$

for $i = 0, 1, \cdots, n$. High pnd times, where the UAV's observability to the radar systems is minimized, scattered with the short periods of low pnd times (high observability regions) as shown in Fig. (2) might make it possible to find optimal trajectories among the radar regions and as well as in the radar regions. This is achieved by driving the radar systems into a condition called *lock–loss* in which a radar system aborts its plans after a specified short time of no detection.

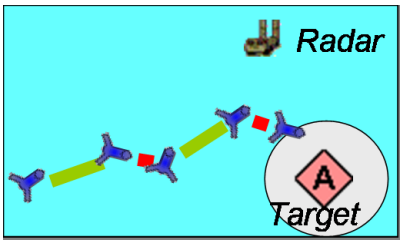

Fig. 2. Two event types; High and Low pnd Periods.

T represents the unknown final event times which is introduced to be able to include temporal constraints in the NTG algorithm. Thus, in the optimization formulations below scaled time variable $\tau = \frac{t}{T}$ is used [4]. The system dynamics for this problem consist of the vehicle dynamics in (4) together with the following dynamics on the new state variables

$$\frac{dT_{2i}}{d\tau} = 0 \quad \text{and} \quad \frac{dT_{2i+1}}{d\tau} = 0 \quad i = 0, ...n-1 \quad (9)$$

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 probability of not–being–detected function, pnd, as;

$$pnd = \prod_{k=1}^{N_r} (1 - pd[k]) \quad (10)$$

where N_r represents number of radar systems. $pd = f_{pd}(sig, rg)$ indicates the probability of detection as a function of signature (sig) and range (rg) values. $sig = f_{sig}(el, az)$ represents the signature and it is a function of elevation (el) and azimuth (az) angles. These two functions are found using tensor product B–Spline functions to fit analytical models to the sig and pd tables shown in Fig. (1) and it will be explained in more details in the next section. After defining these functions, a set of cost and constraint functions can be developed.

B. Cost Functions

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

- Initial Cost Function:

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

- Trajectory Cost Function:

$$J_{tcf} = \int_0^1 \left(\frac{W_u}{T^2}\left(\left(\frac{dn_{ac}}{d\tau}\right)^2 + \left(\frac{de_{ac}}{d\tau}\right)^2\right) - W_p pnd\right) T d\tau$$

W_t, W_u and W_p represent the weight functions on the total mission time, speed and pnd functions, respectively. J_{icf} minimizes the total mission time while J_{tcf} puts penalties on the speed to bound the control action, and on the probability of not–being–detected function to maximize

the survivability of the aircraft. The integral in J_{tcf} is respect to the scaled time, τ, and has bounds from zero to one. 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.

C. Constraint Functions

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

- Linear Initial Constraints:

$$n_{ac}(0) \leq n_{ac}(\tau)|_{\tau=0} \leq n_{ac}(0)$$

$$e_{ac}(0) \leq e_{ac}(\tau)|_{\tau=0} \leq e_{ac}(0)$$

$$\dot{n}_{ac}(0)T \leq \frac{dn_{ac}}{d\tau}|\tau = 0 \leq \dot{n}_{ac}(0)T$$

$$\dot{n}_{ac}(0)T \leq \frac{de_{ac}}{d\tau}|\tau = 0 \leq \dot{n}_{ac}(0)T$$

$$\underline{T_{2i}} \leq T_{2i}|_{\tau=0} \leq \overline{T_{2i}}, \quad i = 0, 1, \cdots n$$

$$\underline{T_{2i+1}} \leq T_{2i+1}|_{\tau=0} \leq \overline{T_{2i+1}}, \quad i = 0, 1, \cdots n$$

where $\dot{(\cdot)} = \frac{d(\cdot)}{dt}$. Since we would like to drive radar systems into the *lockloss* condition $\overline{T_{2i+1}} << \overline{T_{2i}}$ 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.

- Nonlinear Trajectory Constraints:

$$\frac{\underline{v}^2}{\overline{v}^2} \leq W_v \frac{1}{T^2}\left(\left(\frac{dn_{ac}}{d\tau}\right)^2 + \left(\frac{de_{ac}}{d\tau}\right)^2\right) \leq 1$$

$$-1 \leq W_c \frac{\frac{dn_{ac}}{d\tau}\frac{d^2e_{ac}}{d\tau^2} - \frac{de_{ac}}{d\tau}\frac{d^2n_{ac}}{d\tau^2}}{\left(\left(\frac{dn_{ac}}{d\tau}\right)^2 + \left(\frac{de_{ac}}{d\tau}\right)^2\right)^{1.5}} \leq 1$$

$$0 \leq W_r \cdot (n_{ac} - n_{ref})^2 + (e_{ac} - e_{ref})^2 \leq 1$$

$$0 \leq W_s(max(sign_i)) \leq 1, \quad i = 1, 2, \cdots, N_r$$

The first constraint puts a limit on the velocity of the UAV. 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 a straight–line reference trajectory between waypoints within a specified radius so that it can change its trajectory to maximize its pnd function. 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), while during the high pnd region $W_s > 1.0$ (low signature is forced), i.e.;

$$\text{If} \quad T_{2i} < t \leq T_{2i+1} \longrightarrow W_s = 1.0$$

$$\text{If} \quad T_{2i-1} < t \leq T_{2i} \longrightarrow W_s > 1.0$$

Note that the constraint equations are scaled to aid in convergence of the nonlinear optimization codes.

- Nonlinear Final Constraints:

$$0 \leq W_f \left((n_{ac}(1) - n_f)^2 + (e_{ac}(1) - e_f)\right)^2 \leq 1$$

$$\underline{\Theta} \leq W_d \cdot atan\left(\frac{\frac{de_{ac}}{d\tau}}{\frac{dn_{ac}}{d\tau}}\right) \leq \overline{\Theta}$$

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 $\tau = \frac{t}{T} \rightarrow \tau = 1$. 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 Θ is the heading angle between $waypoint(k+1)$ and $waypoint(k)$. $\underline{\Theta}$ and $\overline{\Theta}$ represent lower and upper bounds on the heading angle, respectively.

D. B–Spline Models

Analytical models to the signature (sig) and the probability of detection (pd) tables shown in Fig. (1) can be found efficiently using tensor product B–Spline functions:

$$sig = f_{sig}(el, az) = \sum_{i=1}^{m}\sum_{j=1}^{n} B_{i,k_{el}}(el)B_{j,k_{az}}(az)a_{ij}$$

$$pd = f_{pd}(sig, rg) = \sum_{i=1}^{p}\sum_{j=1}^{r} B_{i,k_{sig}}(sig)B_{j,k_{rg}}(rg)b_{ij}$$

It is important to notice that when azimuth angle changes between ∓ 30 degrees and ∓ 31 degrees there is a big change in the magnitude of the signature function of the UAV as it is shown in Fig. (3). The same phenomena occurs when the azimuth angle changes between ∓ 159 degrees and ∓ 160 degrees. This introduces several local minima and sharp changes in the gradients. Thus, the underlying optimization problem is in fact quite difficult.

Fig. (5) and Fig. (6) show the result of the fit function models by B–spline tensor product functions versus the actual data points indicated by 'o'. The order of the polynomials in the models used are $kel = 4,\ kaz = 2,\ ksig = 2$ and $krg = 4$ and number of coefficients are $m = 9,\ n = 11,\ p = 7$ and $r = 30$. Note the sharp changes in signature for azimuths around ∓ 30 and ∓ 159 degrees in Fig. (5).

el/az	0	+/-30	+/-31	+/-159	+/-160	+/-180
+90	1E0	1E0	1E0	1E0	1E0	1E0
+45	5E-3	5E-3	1E0	1E0	5E-3	5E-3
+20	5E-4	5E-4	5E-1	5E-1	5E-4	5E-4
0	5E-5	5E-5	5E-1	5E-1	5E-5	5E-5
-20	5E-4	5E-4	5E-1	5E-1	5E-4	5E-4
-45	5E-3	5E-3	1E0	1E0	5E-3	5E-3
-90	1E0	1E0	1E0	1E0	1E0	1E0

Fig. 3. Large UAV Signature Data.

Signature Value	Paq = 0.99	Paq = 0.5	Paq = 0.1	Paq = 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

Fig. 4. Large UAV / Medium SAM Probability of Detection (Acquisition) Data.

E. Implementation Issues

Tensor Product B–Spline models can be efficiently found using MATLAB's Spline Toolbox, especially with *spap2* and *spaps* functions. Once the models are found *spbrk* function is used to extract the knots vectors, degree of polynomials and coefficient matrix. Implementing the B–Spline models in the NTG formulations requires little care. The models can be implemented in NTG by writing a subroutine which calculates the B–Spline basis functions and evaluates the value of the models and their derivatives respect to a given location of the aircraft. Interested readers are directed to the information given in the section (II-A) and the book by Carl de Boor [11]. Implementation of derivatives of the B–Spline models, for example for signature function, should be done using the chain rule

$$
\begin{aligned}
el &= f_{el}(\psi(\dot{n}_{ac}, \dot{e}_{ac}), n_{ac}, e_{ac}) \\
az &= f_{az}(\psi(\dot{n}_{ac}, \dot{e}_{ac}), n_{ac}, e_{ac})
\end{aligned}
$$

$$
\begin{aligned}
\frac{dsig}{dn_{ac}} &= \frac{dsig}{del} \cdot \frac{del}{dn_{ac}} + \frac{dsig}{daz} \cdot \frac{daz}{dn_{ac}} \\
\vdots \quad & \quad \vdots \qquad\qquad\qquad \vdots \\
\frac{dsig}{d\dot{e}_{ac}} &= \frac{dsig}{del} \cdot \frac{del}{d\dot{e}_{ac}} + \frac{dsig}{daz} \cdot \frac{daz}{d\dot{e}_{ac}}
\end{aligned}
$$

Derivatives of the pd and hence pnd function can be implemented similarly.

IV. NUMERICAL SIMULATIONS

In order to investigate the proposed approach, several examples are presented in this section. We assume that pitch and bank angles are zero and altitude is fixed at 12km. There are 6 temporal events used between each waypoints in all examples, 3 for high pnd regions, and 3 for low pnd regions. It is important to mention that the radar regions are *not* modeled as obstacles in any examples. The only thing it is assumed 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, shown with circular regions in the examples, the signature and probability of detection are assumed to be zero.

The first simulation considers planning a trajectory from the base location at (-50km , -50km) to a final destination at (150km , 150km). As it is seen in Fig. (7), the resulting trajectory successfully avoids the radar detection systems represented with circular regions in the figure. A straight–line trajectory would directly go through the centers of the radar detection systems in this simulation. Fig. (8) and Fig.(9) show the second simulation. First figure shows the trajectory found through the radar regions between locations (-100km , -40km) and (-20km , 0km). The second one, shows a different trajectory between locations (-100km , -40km) and (-50km , 30km) for the same radar regions as in

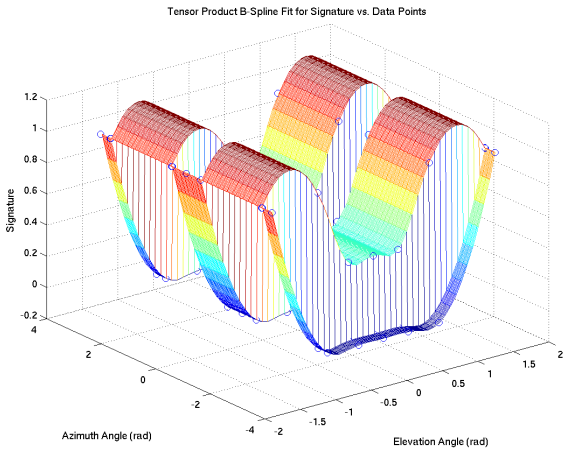

Fig. 5. The Signature Model Approximation Using B–Spline Tensor Product Functions.

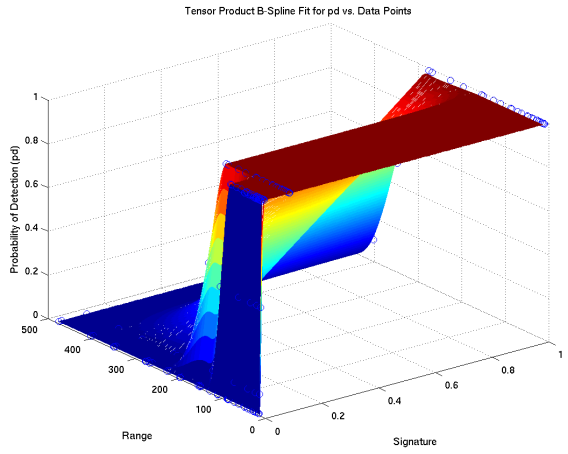

Fig. 6. The pd Model Approximation Using B–Spline Tensor Product Functions.

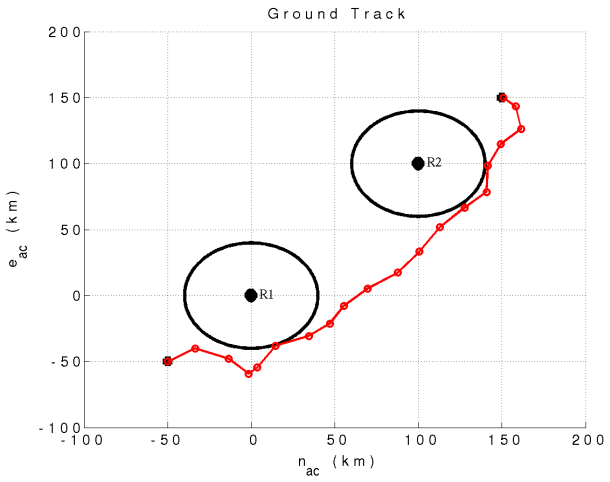

Fig. 7. Ground track of the UAV for 2–radar example.

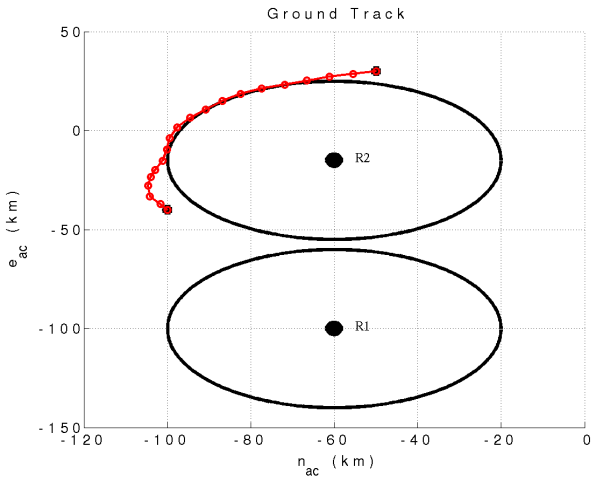

Fig. 9. Ground track of the UAV for 2–radar example.

the first one. Note that this is the shortest path avoiding the radar region since the optimization also tries to minimize the total flight time. A similar simulation is shown in Fig. (10) and Fig. (11) between locations from (-100km , -40km) to (-20km , -120km) and from (-100km , -40km) to (-50km , -150km).

The last simulation is to find an optimal trajectory from a base location, (-100km , -100km) to 3 different target sites located inside radar zones at (0km , 30km), (200km , 20km) and (200km , -130km). Other initial waypoints are located at (-46km , -30km), (100km , 25km) and (200km , -40km). These initial course waypoints construct initial straight–line trajectories between the base point and 1st target, 1st and 2nd targets and finally 2nd and the last targets. Radars are located at (0km , 0km), (200km , 50km) and (200km , -100km). Fig. (12) clearly illustrates that the UAV finds optimum entrance points to the targets to maximize its pnd function and to minimize the total flight time.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a method of finding real–time nonlinear trajectories for unmanned aerial vehicles to maximize the probability of–not–being detected function in minimal total flight time so that survivability of the aircraft is maximized against radar detection systems. This method builds on our initial results previously presented in CDC 2003.

Based on tensor product B–Spline functions, analytical models of the signature and probability of detection tables were found. Then, the trajectory generation problem and its implementation in the NTG were explained in detail. Illustrative numerical simulations which indicate the ability of our proposed method to find real–time nonlinear optimal trajectories from a start point to a final destination point between radar detection systems, as well as, inside radar regions were provided.

We noticed that as the complexity of the problem in-

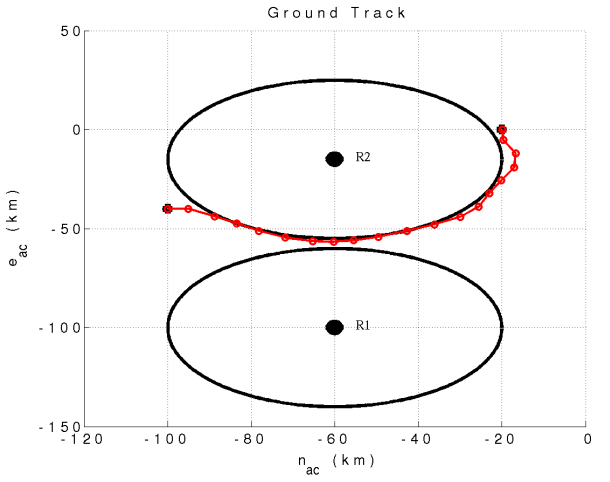

Fig. 8. Ground track of the UAV for 2–radar example.

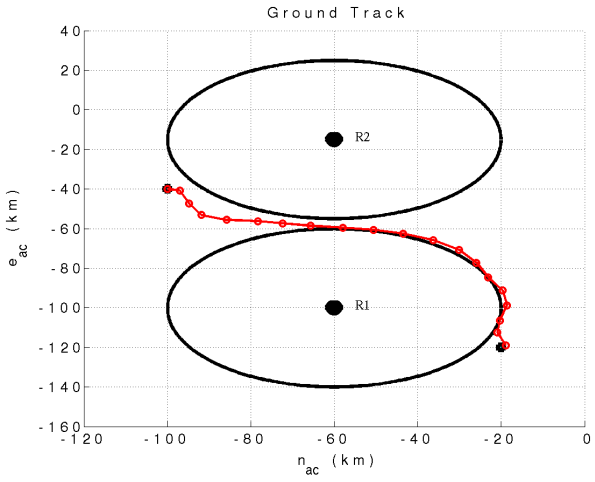

Fig. 10. Ground track of the UAV for 2–radar example.

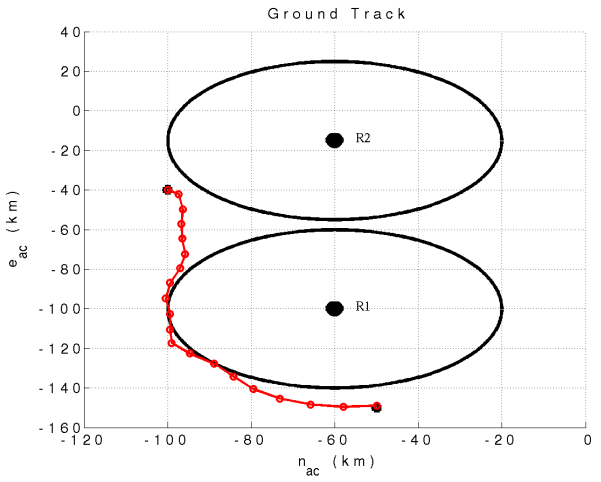

Fig. 11. Ground track of the UAV for 2–radar example.

creases by using probability of detection table with the signature table, this leads to more complicated convergence issues. Our future work will focus on convergence issues, on extending the proposed method to 3D, and on understanding the issues involved with using NTG in a closed loop, receding horizon manner.

REFERENCES

[1] Mark B. Milam, *Real–Time Optimal Trajectory Generation for Constrained Dynamical Systems*, PhD thesis, Control and Dynamical Systems, Caltech, 2003.

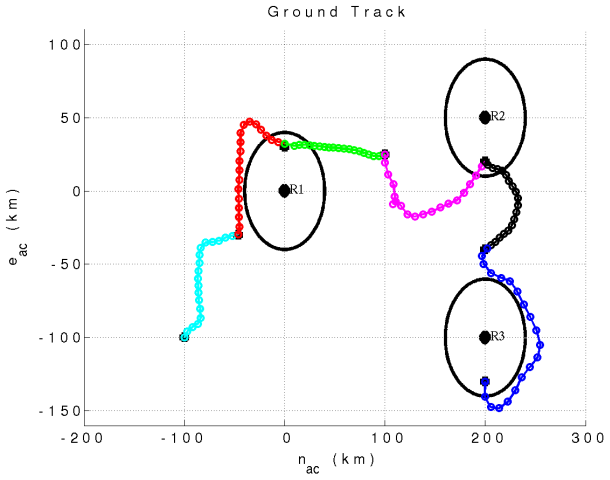

Fig. 12. Ground track of the UAV for 3–radar example.

[2] Milam M.B., Missile Interception Research Report, *California Institute of Technology Internal Report*, 2002.

[3] Milam M.B. and Mushambi K. and Murray R.M., "A Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems", *Conference on Decision and Control*, 2000.

[4] Milam M.B. and Franz R. and Murray R.M., "Real-Time Constrained Trajectory Generation Applied to a Flight Control Experiment", *IFAC*, 2002.

[5] Lian F.-L. and Murray R.M., "Cooperative Task Planning of Multi-Robot Systems with Temporal Constraints", *International Conference on Robotics and Automation*, 2003.

[6] Lian F.-L. and Murray R.M., "Real-Time Trajectory Generation for the Cooperative Path Planning of Multi-Vehicle Systems", *Conference on Decision and Control*, 2002.

[7] Corman D. and Knutti J., *MICA OEP 1.1 User Guide*, The Boeing Company Internal Report, 2003.

[8] Gill P.E. and Murray W. and Saunders M.A. and Wright M. H., *User's Guide for NPSOL 5.0 A Fortran Package for Nonlinear Programming*, Systems Optimization Laboratory, Stanford University, Stanford CA, 1998.

[9] C. S. Lin and P.R. Chang and J. Y. S. Luh, Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots, *IEEE Transactions on Automatic Control*, v. AC-28, n. 12, pg. 1066-1074, December, 1983.

[10] Z. Shiller and Y. R. Gwo, Dynamic Motion Planning of Autonomous Vehicles, *IEEE Transactions on Robotics and Automation*, v. 7, n. 2, pg. 241-249, April, 1991.

[11] Carl de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 2001.

[12] C. P. Neuman and A. Sen, A Suboptimal Control Algorithm for Constrained Problems using Cubic Splines, *Automatica*, v. 9, pg. 601-613, 1973.

[13] John T. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*, Society of Industrial and Applied Mathematics, 2001.

[14] Murray R.M. and Milam M.B., *NTG - a library for real-time trajectory generation*, 2002.

[15] McFarland M.B. and Acjeru R.A. and Taylor B.K., "Motion Planning for Reduced Observability of Autonomous Aerial Vehicles", *IEEE International Conference on Control Applications*, 1999.

[16] Fliess M. and Levine J. and Martin P. and Rouchon P., Flatness and Defect of Nonlinear Systems: Introductory Theory and Examples, *International Journal of Control*, (61)6 1327-1360, 1995.

[17] Jongrae Kim and João P. Hespanha, "Discrete Approximation to Continuous Shortest–Path: Application to Minimum–Risk Path Planning for Groups of UAVs", *42nd IEEE Conference on Decision and Control, Maui, Hawaii, 9–12 December*, 2003.

[18] Kathy Misovec, Tamer Inanc, Jerry Wohletz and Richard M. Murray, "Low–Observable Nonlinear Trajectory Generation for Unmanned Air Vehicles", *42nd IEEE Conference on Decision and Control, Maui, Hawaii, 9–12 December*, 2003.

[19] Martin Norsell, "Radar Cross Section Constraints in Flight Path Optimization", *41st Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 6–9 January, 2003.

[20] Michael C. Novy, David R. Jacques and Meir Pachter, "Air Vehicle Optimal Trajectories Between Two Radars", *Proceedings of the American Control Conference*, Anchorage, AK, May 8–10, 2002.

[21] Meir Pachter and Feffrey Hebert, "Optimal Aircraft Trajectories for Radar Exposure Minimization", *Proceedings of the American Control Conference*, Arlington, VA, June 25–27, 2001.