A Formally Verified Conflict Detection Algorithm for Polynomial Trajectories

Anthony Narkawicz, César Muñoz*

NASA, Langley Research Center, Hampton, VA, 23681, USA

In air traffic management, conflict detection algorithms are used to determine whether or not aircraft are predicted to lose horizontal and vertical separation minima within a time interval assuming a trajectory model. In the case of linear trajectories, conflict detection algorithms have been proposed that are both sound, i.e., they detect all conflicts, and complete, i.e., they do not present false alarms. In general, for arbitrary nonlinear trajectory models, it is possible to define detection algorithms that are either sound or complete, but not both. This paper considers the case of nonlinear aircraft trajectory models based on polynomial functions. In particular, it proposes a conflict detection algorithm that precisely determines whether, given a lookahead time, two aircraft flying polynomial trajectories are in conflict. That is, it has been formally verified that, assuming that the aircraft trajectories are modeled as polynomial functions, the proposed algorithm is both sound and complete.

I. Introduction

Separation requirements in the airspace are typically given by a minimum horizontal separation, e.g., 5 nautical miles, and a minimum vertical separation, e.g., 1000 feet.[13] A loss of separation between two aircraft occurs when both of these minima are simultaneously violated, and a conflict occurs when the aircraft are predicted to lose separation in the near future, usually 5 minutes. Conflict detection algorithms have as input the state information of two aircraft and a lookahead time. They return a Boolean value indicating whether or not the aircraft are in conflict, i.e., they are predicted to be in a loss of separation within the given lookahead time. When a conflict is detected, conflict resolution algorithms compute resolution maneuvers for the aircraft that maintain the required aircraft separation. Conflict detection and resolution (CD&R) systems are part of computer-based systems that assist pilots and air traffic controllers to maintain safety in the airspace by keeping aircraft separated. These separation assurance systems are critical elements of air/ground distributed operational concepts for the next generation of air traffic management systems such as the US's Next Generation of Air Traffic Systems (NGATS)[22] and Europe's Single European Sky ATM Research (SESAR).[a]

CD&R algorithms rely on the reported state information of the aircraft. This state information typically includes 3D position and velocity vectors. A given aircraft trajectory model is then used to propagate the current state information forward in the future within the time interval determined by the lookahead time. Several state propagation methods for CD&R systems have been proposed.[8] For example, state-based conflict detection algorithms use a linear projection of the current state of the aircraft. This simple aircraft trajectory model corresponds to a point mass moving along a straight line at constant speed. More sophisticated state propagation methods assume nonlinear trajectories or probabilistic trajectory models.

Three important safety properties for conflict detection algorithms are soundness, completeness, and correctness. Given an aircraft trajectory model, an algorithm is sound if it only detects potential conflicts, i.e., if in every situation where the algorithm returns `true`, the aircraft are in conflict according to the trajectory model, then the detection algorithm is sound. An algorithm is complete if all conflicts are detected, i.e., if in every situation where two aircraft are in conflict according to the trajectory model, the algorithm returns `true`, then the detection algorithm is complete. Finally, a detection algorithm is correct if it is both

[*] Research Computer Scientist, AIAA Member.

[a] `http://www.eurocontrol.int/content/sesar-and-research.`

sound and complete, meaning that the algorithm returns `true` if and only if the aircraft's trajectories are in conflict. The notions of soundness and completeness are related to the notions of false alerts and missed alerts and they may play a role in the development of safety cases for the certification of CD&R systems.

For linear trajectories, i.e., trajectories where the initial velocity does not change within the lookahead time, it is possible to define algorithms that are correct, i.e., sound and complete.[3, 9] Unfortunately, for nonlinear trajectory models, designing a conflict detection algorithm that is correct is more challenging. One way to design a detection algorithm for an arbitrary trajectory model is to test a number of sample points, representing aircraft positions within a given lookahead time, and return the Boolean value `true` if some of those points are in loss of separation. Such an algorithm is sound but not complete, since it cannot detect conflicts that occur outside the set of sample points.

In previous work, the authors proposed a detection algorithm for arbitrary nonlinear trajectory models and formally verified its main safety properties.[11] That algorithm is based on a numerical method using Bernstein polynomials, which are a particular case of spline functions. The algorithm explicitly computes a small interval enclosure for the smallest distance between two aircraft during a lookahead time, and returns a Boolean value depending on this information. That algorithm can be proved to be correct within some approximation bounds. More precisely, by modifying the separation minima (both horizontal and vertical), the algorithm is provably sound or provably complete. However, for given separation minima it cannot simultaneously satisfy both properties.

In this paper, the authors present a new, formally verified conflict detection algorithm for aircraft trajectories described by polynomials in the time variable. This algorithm is provably correct. Thus, given the state information of two aircraft and a lookahead time, it returns the Boolean value `true` if and only if the aircraft, which are assumed to fly polynomial trajectories, are predicted to be in loss of separation within the lookahead time. The proposed algorithm is based on a well-known result in real algebraic geometry called *Tarski's theorem.* This theorem enables the computation of a Boolean value that precisely determines whether or not the distance between two polynomial trajectories ever crosses a certain separation threshold within a time interval. In the case of linear trajectories, the quadratic formula can be used to determine whether a polynomial of degree 2, i.e., the square of the distance between two aircraft at any time, ever crosses the separation minima. In the case of polynomial trajectories of higher degree, Tarski's theorem can be used to make the same determination.

The rest of the paper is organized as follows. The conflict detection problem is discussed in Section II. Tarski's theorem is described in Section III. This theorem is the backbone of the conflict detection algorithm for polynomial trajectories that is proposed in Section IV. The last section discusses related work and concludes the paper. The proposed conflict detection algorithm and its correctness property have been *formally* specified and verified in the Prototype Verification System (PVS).[14] To make this paper accessible to non-PVS users, this paper uses mathematical notation instead of PVS concrete syntax.

II. Conflict Detection

Since conflicts between multiple aircraft can be detected in a pairwise fashion, this paper only considers conflicts between two aircraft. These two aircraft are referred to as the *ownship* and the *intruder.* As usual in CD&R literature, the airspace volume is modeled using a flat-earth projection in a 3-dimensional rectangular coordinate system. That is, aircraft positions are viewed as points in $\mathbb{R}^3$. The separation requirement between two aircraft is specified as a minimum horizontal separation D and a minimum vertical separation H. Typically, D is 5 nautical miles and H is 1000 feet.[13] In this paper, D and H are considered to be known numerical constants. The separation requirement can be understood as an imaginary horizontal cylinder, called the *protected zone*, of height $2H$ and diameter $2D$ around the intruder aircraft.

A loss of separation between the ownship and the intruder aircraft occurs when the horizontal distance between the aircraft is less than D and the vertical distance is less than H, i.e., when the ownship is in the interior of the intruder's protected zone. Let $\mathbf{s}_o \in \mathbb{R}^3$ and $\mathbf{s}_i \in \mathbb{R}^3$ be the current positions of the ownship and intruder aircraft, respectively. Formally, the ownship and intruder aircraft are said to be in *loss of separation* if the following predicate on $\mathbf{s}_o$ and $\mathbf{s}_i$, holds.

$$\mathit{los?}(\mathbf{s}_o, \mathbf{s}_i) \equiv |s_z| < H \;\text{ and }\; \|\mathbf{s}_{(x,y)}\| < D,$$

where $\mathbf{s} = \mathbf{s}_o - \mathbf{s}_i$, i.e., $\mathbf{s}$ is the relative position of the ownship with respect to the intruder aircraft, and $\mathbf{s}_{(x,y)}$ is the horizontal projection of 3-dimensional vector $\mathbf{s}$.

II.A. Trajectories

An aircraft trajectory represents the set of possible positions for the aircraft according to some state propagation model.[8] A state propagation model for CD&R systems may be as simple as a linear projection of the current position at the current constant velocity. More complicated models consider uncertainties in the aircraft state due to aircraft dynamics, weather patterns, and other factors. In this paper, an *aircraft trajectory* is a continuous function that maps a time in $\mathbb{R}$ to an aircraft position in $\mathbb{R}^3$. Given a time $t \in \mathbb{R}$, the evaluation of a trajectory at time t is a point in $\mathbb{R}^3$ that represents the projected 3-dimensional position for the aircraft at the time t.

Example 1 (Linear Dynamics)**.** *Tactical state-based CD&R systems uses an aircraft trajectory model that assumes a linear projection of its current position* $\mathbf{s} \in \mathbb{R}^3$ *along its current velocity* $\mathbf{v} \in \mathbb{R}^3$*. This type of trajectory can be represented by the parametric function* $\boldsymbol{linear}_{\mathbf{s},\mathbf{v}} : \mathbb{R} \to \mathbb{R}^3$*, with parameters* $\mathbf{s}$ *and* $\mathbf{v}$*, defined by*

$$\boldsymbol{linear}_{\mathbf{s},\mathbf{v}}(t) \equiv \mathbf{s} + t\ \mathbf{v}. \tag{1}$$

Example 2 (Turn Dynamics)**.** *During a steady coordinated turn without friction, the position of an aircraft will follow a circle of radius* $\frac{\nu^2}{g\ \tan\phi}$*, where* ν *is the true air speed,* g *is the acceleration of gravity, and* ϕ *is the bank angle of the aircraft. Thus, the trajectory of an aircraft during a turn can be represented by the parametric function* $\boldsymbol{turn}_{\mathbf{s},r,\alpha,\omega,v_z} : \mathbb{R} \to \mathbb{R}$*, with parameters* $\mathbf{s}$*,* r*,* α*,* ω*, and* v_z*, defined by*

$$\boldsymbol{turn}_{\mathbf{s},r,\alpha,\omega,v_z}(t) \equiv \mathbf{s} + (r\sin(\alpha + t\,\omega), r\cos(\alpha + t\,\omega), t\,v_z), \tag{2}$$

where $\mathbf{s}$ *is the center point of the turn,* $\omega = \pm\frac{g}{\nu}\tan\phi$*,* α *is the angle along the turn at time zero,* $r = \frac{\nu^2}{g\ \tan\phi}$*, and* v_z *is the vertical speed.*

Henceforth, trajectories for the ownship and intruder aircraft are denoted by P_o and P_i, respectively. Specifically, trajectories will be studied where each of the components functions of P_o and P_i are defined with polynomials in a time variable t.

II.B. Conflict Detection Algorithms

While loss of separation is formalized as a predicate on two aircraft positions $\mathbf{s}_o$ and $\mathbf{s}_i$, a conflict between two aircraft is formalized as a predicate on the ownship and intruder trajectories P_o and P_i in $\mathbb{R} \to \mathbb{R}^3$, respectively. The conflict predicate is defined for a lookahead time T that represents a time interval of interest. As in the case of D and H, T is assumed to be a known numerical constant. The trajectories P_o and P_i are in *conflict* if there exists $t \in [0, T]$ such that the positions $P_o(t)$ and $P_i(t)$ are in loss of separation:

$$\textit{conflict?}(P_o, P_i) \equiv \exists\, t \in [0, T] : \textit{los?}(P_o(t), P_i(t)). \tag{3}$$

Example 3. *If both trajectories,* P_o *and* P_i*, are given by linear projections of the states of the aircraft at time zero, then* $P_o(t) = \mathbf{s}_o + t\,\mathbf{v}_o$ *and* $P_i(t) = \mathbf{s}_i + t\,\mathbf{v}_i$*, where* $\mathbf{s}_o$*,* $\mathbf{s}_i$*,* $\mathbf{v}_o$*, and* $\mathbf{v}_i$ *are the positions and velocities of the ownship and the intruder at time zero, respectively. In this case,*

$$\begin{aligned}\textit{conflict?}(P_o, P_i) \iff \exists\, t \in [0, T] : &|\mathbf{s}_z + t\,\mathbf{v}_z| < H \ \ \textit{and} \\ &\|\mathbf{s}_{(x,y)} + t\,\mathbf{v}_{(x,y)}\| < D,\end{aligned}$$

where $\mathbf{s} = \mathbf{s}_o - \mathbf{s}_i$ *and* $\mathbf{v} = \mathbf{v}_o - \mathbf{v}_i$*. This definition is typically used in state-based CD&R.*[3, 9]

An algorithm used by an aircraft to detect conflicts with another aircraft is called a *conflict detection algorithm.* In this paper, a conflict detection algorithm is a function `cd` that takes as inputs P_o and P_i, and returns a Boolean value. Formally, a conflict detection algorithm `cd` is *complete* if for all trajectories P_o, P_i such that *conflict?*$(P_o, P_i) =$ `true`, it holds that `cd`$(P_o, P_i) =$ `true`. Similarly, it is *sound* if for trajectories P_o, P_i such that `cd`$(P_o, P_i) =$ `true`, it holds that *conflict?*$(P_o, P_i) =$ `true`. Finally, the algorithm `cd` is *correct* if it is both sound and complete.

II.C. Conflict For Polynomial Trajectories

In this paper, a state propagation model based on polynomial trajectories is considered. That is, it is assumed that

$$
\begin{aligned}
P_o(t) &\equiv (a_q t^q + \cdots + a_1 t + a_0,\;\; b_r t^r + \cdots + b_1 t + b_0,\;\; c_s t^s + \cdots + c_1 t + c_0),\\
P_i(t) &\equiv (d_k t^k + \cdots + d_1 t + d_0,\;\; e_l t^l + \cdots + e_1 t + e_0,\;\; f_m t^m + \cdots + f_1 t + f_0).
\end{aligned}
$$

where q, r, s, k, l, and m are, respectively, the degrees of the polynomials appearing above.

Given the explicit descriptions above of these trajectories, conflict between these trajectories can be specified as follows.

$$
\begin{aligned}
&\mathit{conflict?}(P_o, P_i) \;\equiv\\
&\quad \exists\, t \in \mathbb{R} :\\
&\qquad t \,\geq\, 0 \;\text{ and}\\
&\qquad T - t \,\geq\, 0 \;\text{ and}\\
&\qquad D^2 \,-\, ((a_q t^q + \cdots + a_0) - (d_k t^k + \cdots + d_0))^2 \,-\\
&\qquad\qquad ((b_r t^r + \cdots + b_0) - (e_l t^l + \cdots + e_0))^2 \,> 0 \;\text{ and}\\
&\qquad H^2 \,-\, ((c_s t^s + \cdots + c_0) - (f_m t^m + \cdots + f_0))^2 \,> 0.
\end{aligned}
\tag{4}
$$

Thus, detecting a conflict for these polynomial trajectories is equivalent to solving this system of four polynomial relations. Indeed, each of the last four lines of the formula is of the form $p(t)\,R\,0$, where R is in the set $\{\geq, >\}$, and where $p(t)$ is a polynomial in the variable t.

For a linear trajectory model, Formula 4 can be reduced to

$$
\begin{aligned}
&\mathit{conflict?}(P_o, P_i) \;\equiv\\
&\quad \exists\, t \in \mathbb{R} :\\
&\qquad t \,\geq\, 0 \;\text{ and}\\
&\qquad T - t \,\geq\, 0 \;\text{ and}\\
&\qquad D^2 \,-\, ((v_{ox} t + s_{ox}) - (v_{ix} t + s_{ix}))^2 \,-\, ((v_{oy} t + s_{oy}) - (v_{iy} t + s_{iy}))^2 \,> 0 \;\text{ and}\\
&\qquad H^2 \,-\, ((v_{oz} t + s_{oz}) - (v_{iz} t + s_{iz}))^2 \,> 0,
\end{aligned}
$$

where $P_o \equiv (s_{ox} + tv_{ox}, s_{oy} + tv_{oy}, s_{oz} + tv_{oz})$, $P_i \equiv (s_{ix} + tv_{ix}, s_{iy} + tv_{iy}, s_{iz} + tv_{iz})$, and $\mathbf{s}_o, \mathbf{s}_i, \mathbf{v}_o, \mathbf{v}_i$ are the positions and velocities of the ownship and the intruder at time zero.

III. Tarski's Theorem

In Section II.C, it is shown that the problem of detecting conflicts for polynomial trajectories is equivalent to determining whether a system of four polynomial equations has a solution t, where t is a real number. There is an algorithm that can efficiently determine whether or not this system of polynomials has a solution. Such an algorithm belongs to the mathematics field of semi-algebraic geometry,[1] which is the study of systems of polynomial relations. The algorithm presented in this paper is a particular instance of a more general algorithm for determining the existence of solutions of any system of polynomial relations.

To illustrate how it is possible to analytically determine whether a polynomial relation has a solution, consider first the simple case of a single quadratic polynomial inequality $at^2 + bt + c \leq 0$, where $a > 0$. This quadratic opens upward, and therefore this equation has a solution if and only if there exists at least one root of this polynomial, meaning that there exists some t where $at^2 + bt + c = 0$. However, by using the quadratic equation, it is relatively easy to see that this happens if and only if $b^2 - 4ac \geq 0$. Thus, the analytic way to check whether $at^2 + bt + c \leq 0$ has a solution is to check whether $b^2 - 4ac \geq 0$. This shows that determining analytically whether a polynomial formula has a solution is possible, at least in the case where the polynomial is a quadratic.

In fact, it is possible to determine analytically whether *any* polynomial system has a solution. The algorithm used in this paper is based on *Tarski's theorem.* First, recall that the extended real numbers $\mathbb{R}^*$ are defined as the real numbers $\mathbb{R}$ with two extra points added, namely ∞ and $-\infty$. Any polynomial p can be evaluated at any point of $\mathbb{R}^*$, and it returns another extended real number in $\mathbb{R}^*$. For instance, if p is the polynomial $p(t) = t^2$, then $p(\infty) = \infty$ and $p(-\infty) = \infty$, and if p is the polynomial $p(t) = -t^3$, then $p(\infty) = -\infty$ and $p(-\infty) = \infty$. Next, let g and h be univariate polynomials, such that h is nonzero. Using the standard Euclidean division algorithm for polynomials, it is always possible to find polynomials q and r such that $g = q \cdot h + r$ and the degree of r is less than the degree of h. Let $\mathrm{rem}(g, h)$ denote the polynomial r after division, known as the *remainder*. Given univariate polynomials p and g, the *Sturm sequence* of p and g is a sequence S of polynomials

$$p_0,\ p_1,\ p_2,\ \dots, p_m, \tag{5}$$

where

$$\begin{aligned} p_0 &= p, \\ p_1 &= g \cdot p', \\ \forall\, d > 1\ : p_d &= -\mathrm{rem}(p_{d-2}, p_{d-1}), \\ p_m &= 0, \text{and} \\ p_{m-1} &\neq 0. \end{aligned} \tag{6}$$

Evaluating each of the polynomials in a Sturm sequence at some $x \in \mathbb{R}^*$ produces a sequence of extended real numbers. A function $\sigma_{p,g}$ is defined on $\mathbb{R}^*$ by setting $\sigma_{p,g}(x)$ to be equal to the number of sign changes in this sequence. When counting the number of sign changes in an evaluated Sturm sequence, any zeros are ignored. For example, if $m = 7$ and $p_0(x) = 4$, $p_1(x) = -3$, $p_2(x) = -5$, $p_3(x) = 0$, $p_4(x) = 18$, $p_5(x) = -4$, $p_6(x) = -1$ and $p_7(x) = 0$, there are sign changes between $p_0(x)$ and $p_1(x)$, between $p_2(x)$ and $p_4(x)$, and between $p_4(x)$ and $p_5(x)$. In this case, the number of sign changes in the sequence is given by $\sigma(x) = 3$.

A basic form of Tarski's theorem states that for $a, b \in \mathbb{R}^*$ with $a < b$, if neither a nor b is a root of both p and $p' \cdot g$, then

$$\begin{aligned} \sigma_{p,g}(a) - \sigma_{p,g}(b) = \mathrm{card}(\{x \in (a, b] :\ p(x) = 0 \ \text{ and }\ g(x) > 0\}) \ - \\ \mathrm{card}(\{x \in (a, b] :\ p(x) = 0 \ \text{ and }\ g(x) < 0\}). \end{aligned}$$

Here, the function $\mathrm{card}(S)$ denotes the cardinality of a finite set S. The case where g is the constant polynomial 1 is commonly known as Sturm's theorem.[21] The basic version of Tarski's theorem motivates the definition of the *Tarski query*, TQ, which is a function with polynomials p and g as inputs.

$$\mathrm{TQ}(p, g) \equiv \sigma_{p,g}(-\infty) - \sigma_{p,g}(\infty).$$

Theorem 1. *Let p, g be univariate polynomials. Then*

$$\mathrm{TQ}(p, g) \ = \ \mathrm{card}(\{x \in \mathbb{R} :\ p(x) = 0 \ \textit{ and }\ g(x) > 0\}) \ - \ \mathrm{card}(\{x \in \mathbb{R} :\ p(x) = 0 \ \textit{ and }\ g(x) < 0\}).$$

The proof of Theorem 1 can be found in works on real algebraic geometry.[1] Theorem 1 is enough to prove the correctness theorem of the conflict detection algorithm presented in this paper. That correctness theorem is Theorem 5. However, we now discuss how, in general Theorem 1 above can be used to solve arbitrary systems of polynomials. However, this general framework, as just noted, is not required to prove the main correctness theorem (Theorem 5).

Well-written expositions of Sturm's and Tarski's theorems can be found in the literature.[1,4,20] Instantiating the polynomial g with 1, g, and g^2 in Theorem 1, it can be seen that the following equality of vectors holds, where there is a matrix multiplication on the right hand side.

$$\begin{bmatrix} \mathrm{TQ}(p, 1) \\ \mathrm{TQ}(p, g) \\ \mathrm{TQ}(p, g^2) \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} \mathrm{card}(S_=) \\ \mathrm{card}(S_>) \\ \mathrm{card}(S_<) \end{bmatrix}, \tag{7}$$

where $S_R = \{x \in \mathbb{R} :\ p(x) = 0 \ \text{ and }\ g(x)\ R\ 0\}$ and

$$\mathbf{M} \equiv \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix}. \tag{8}$$

Since the matrix $\mathbf{M}$ is invertible, the vector on the far right hand side can be computed by calculating the three Tarski queries.

In the following sections, entries of matrices are expressed with indices starting at 0. The top left entry of a matrix is its $(0,0)$-th entry, and the first entry of a vector is its 0-th entry. The expression $\mathbf{M}[i,j]$ denotes the (i,j) entry of a matrix $\mathbf{M}$. Let $\mathsf{g} = \{g_0,\ldots,g_k\}$ be any sequence of polynomials and define $\mathbf{TQ}(p,\mathsf{g})$ to be the vector with 3^{k+1} entries whose i-th entry is given by

$$\mathrm{TQ}(p, \prod_{d=0}^{k} g_d^{i_d}).$$

where $(i_0,\ldots,i_k)$ is the base-3 representation of i. Let $\mathbf{NSol}(p,\mathsf{g})$ be the vector with 3^{k+1} entries whose j-th entry is given by the cardinality of the set

$$\texttt{SolSet}(p,\mathsf{g},j) = \{x \in \mathbb{R} : \ p(x) = 0 \ \text{ and } \ g_0(x)\ R_0\ 0 \ \text{ and } \ \ldots \ \text{ and } \ g_k(x)\ R_k\ 0\},$$

where each relation R_d, with $0 \le d \le k$, is given by

$$R_d \equiv \begin{cases} = & \texttt{if} \ j_d = 0, \\ > & \texttt{if} \ j_d = 1, \\ < & \texttt{if} \ j_d = 2, \end{cases}$$

and where $(j_0,\ldots,j_k)$ is the base-3 representation of j.

Theorem 2. *For any polynomial g and sequence $\mathsf{g} = (g_0,\ldots,g_k)$,, all with real coefficients,*

$$\mathbf{TQ}(p,\mathsf{g}) = \mathbf{M}^{\otimes(k+1)} \cdot \mathbf{NSol}(p,\mathsf{g}). \tag{9}$$

Theorem 2 and its proof can be found in works on real algebraic geometry.[1] The matrix $\mathbf{M}^{\otimes(k+1)}$ in Formula (9) denotes the standard $(k+1)$ tensor power of the matrix $\mathbf{M}$. The matrix $\mathbf{M}^{\otimes(k+1)}$ is invertible and its inverse is given by the following formula.

$$(\mathbf{M}^{\otimes(k+1)})^{-1} = (\mathbf{M}^{-1})^{\otimes(k+1)} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}^{\otimes(k+1)}.$$

The next result following immediately from this

Theorem 3. *For a nonzero polynomial p and a sequence of nonzero polynomials $\mathsf{g} = (g_0,\ldots,g_k)$,, all with real coefficients,*

$$\mathbf{NSol}(p,\mathsf{g}) = (\mathbf{M}^{-1})^{\otimes(k+1)} \cdot \mathbf{TQ}(p,\mathsf{g}).$$

The theorem above follows directly from the discussion above, and a more indepth discussion and proof can be found in works on real algebraic geometry.[1] Theorem 3 enables the effective computation of $\mathbf{NSol}(p,\mathsf{g})$, which are cardinalities of sets of the form

$$\{t \in \mathbb{R} : \ p(t) = 0 \ \text{ and } \ g_0(x)\ R_0\ 0 \ \text{ and } \ \ldots \ \text{ and } \ g_k(x)\ R_k\ 0\},$$

with $R_d \in \{=,>,<,\neq,\geq,\leq\}$ for $0 \le d \le k$. That is, this theorem makes it possible to count solutions to sets of relations, provided that one of the relations is an equality. In the more general case, it is always possible to reduce any system of polynomials with relations in $\{=,>,<,\neq,\geq,\leq\}$, to a system of polynomials where one of the relations is an equality. This can be done by adding one extra polynomial equation, where the polynomial in question is either the product of the polynomials in the system or the derivative of that product. This is stated by the following theorem, whose reasoning follows from standard theorems in real analysis.[1, 19]

Theorem 4. *Consider a collection of polynomials $g_0,\ldots,g_k$ and relations $R_0,\ldots,R_k$, where $R_d \in \{=,>,<,\neq,\geq,\leq\}$ for $0 \le d \le k$. Suppose that the system $S \equiv g_0(t)\ R_0\ 0$ and $\ldots$ and $g_k(t)\ R_k\ 0$ is not satisfied at either $-\infty$ or ∞. Then S has a solution $t \in \mathbb{R}$ if and only if one of the following two conditions holds, where Q is the polynomial $\prod_{d=0}^{k} g_d$.*

- *S and $Q = 0$ are satisfiable at a common point.*
- *S and $Q' = 0$ are satisfiable at a common point.*

IV. Conflict Detection Algorithm for Polynomial Trajectories

Let P_o and P_i be the polynomial trajectories described in Section II.C. Recall from that section that conflict detection between P_o and P_i is equivalent to determining whether the following system of polynomials has a solution $t \in \mathbb{R}$.

$$\mathtt{CDp}_0(t) \geq 0 \text{ and } \mathtt{CDp}_1(t) \geq 0 \text{ and } \mathtt{CDp}_2(t) > 0 \text{ and } \mathtt{CDp}_3(t) > 0,$$

where the polynomials $\mathtt{CDp}_0$, $\mathtt{CDp}_1$, $\mathtt{CDp}_2$, and $\mathtt{CDp}_3$ are defined by

$$\begin{aligned}
\mathtt{CDp}_0(t) &\equiv t,\\
\mathtt{CDp}_1(t) &\equiv -t + T,\\
\mathtt{CDp}_2(t) &\equiv D^2 - ((a_q t^q + \cdots + a_0) - (d_k t^k + \cdots + d_0))^2 - ((b_r t^r + \cdots + b_0) - (e_l t^l + \cdots + e_0))^2,\\
\mathtt{CDp}_3(t) &\equiv H^2 - ((c_s t^s + \cdots + c_0) - (f_m t^m + \cdots + f_0))^2.
\end{aligned}$$

Theorem 4 makes it possible to define a conflict detection algorithm for trajectories of this type by computing the coefficients of the appropriate row of the matrix $(\mathbf{M}^{-1})^{\otimes(k+1)}$ as well as the vector $\mathbf{TQ}(p, \mathsf{g})$. This enables the direct computation of the corresponding element of the vector $\mathbf{NSol}(p, \mathsf{g})$. In fact, the algorithm first simplifies the above system by noting that if either $\mathtt{CDp}_0(t) = 0$ or $\mathtt{CDp}_1(t) = 0$, then this system has a solution at either 0 or T. Thus, the algorithm first checks whether there is a solution at 0 or T and then uses Theorem 4 to check whether there is a solution to the following system, which only includes $>$ relations and no $\geq$ relations.

$$\mathtt{CDp}_0(t) > 0 \text{ and } \mathtt{CDp}_1(t) > 0 \text{ and } \mathtt{CDp}_2(t) > 0 \text{ and } \mathtt{CDp}_3(t) > 0. \tag{10}$$

Theorem 4 implies that the product of one of the polynomials is zero at the solution point, since their product Q is zero at that point. However, the system of polynomial relations in Formula (10) has only $>$ relations, so it is impossible that the product of these polynomials is zero at any point where this system is satisfied. Thus, the only other possibility for the conditions in Theorem 4 to have a solution is for a solution to exist at a point where the derivative of the product of these four polynomials is zero. This motivates the definition of the conflict detection algorithm in Figure 1 for polynomial trajectories P_o and P_i. The algorithm below returns a Boolean value depending on whether the aircraft are in conflict or not.

The sum of 16 Tarski queries that appears in the definition of the algorithm `cd_poly` in Figure 1 is equal to twice the dot product of the 40-th row of $(\mathbf{M}^{\otimes 4})^{-1}$ with the vector $\mathbf{TQ}(\Pi, \{g_0, g_1, g_2, g_3\})$, where, as in the algorithm above, $g_i \equiv \mathtt{CDp}_i$, for $0 \leq i \leq 3$, and $\Pi \equiv g_0 \cdot g_1 \cdot g_2 \cdot g_3$. The 40-th row of this matrix corresponds to the 40-th entry of the vector

$$\mathbf{NSol}(\Pi, \{g_0, g_1, g_2, g_3\}),$$

which is given by the following cardinality:

$$\operatorname{card}(\{t \in \mathbb{R} : \Pi'(t) = 0 \text{ and } g_0(t) > 0 \text{ and } g_1(t) > 0 \text{ and } g_2(t) > 0 \text{ and } g_3(t) > 0\}).$$

The correctness theorem for the algorithm above is presented below. It is the main result of this paper.

Theorem 5 (Correctness for Polynomial Trajectories)**.** *The conflict detection algorithm `cd_poly` is both sound and complete, and therefore also correct, for polynomial trajectories. That is for all polynomial trajectories P_o and P_i, conflict?$(P_o, P_i) =$ `true`, i.e., the trajectories are in conflict, if and only if*

$$\mathtt{cd\_poly}(P_o, P_i) = \mathtt{true}.$$

Theorem 5 states that, assuming a polynomial trajectory model, the algorithm `cd_poly` precisely detects all conflicts, i.e., it does not miss any conflict and it does not return `true` when aircraft trajectories are not actually in conflict.

```
cd_poly$(P_o, P_i) \equiv$
  let
    $g_0 = \texttt{CDp}_0, g_1 = \texttt{CDp}_1, g_2 = \texttt{CDp}_2, g_3 = \texttt{CDp}_3,$
    $\Pi = g_0 \cdot g_1 \cdot g_2 \cdot g_3$ in
  if $\mathit{los?}(P_o(0), P_i(0))$ or $\mathit{los?}(P_o(T), P_i(T))$ then
    true
  elsif $g_0 = 0$ or $g_1 = 0$ or $g_2 = 0$ or $g_3 = 0$ or $\Pi' = 0$ then
    false
  elsif
      $\mathrm{TQ}(\Pi, g_0g_1g_2g_3) + \mathrm{TQ}(\Pi, g_0g_1g_2g_3^2) + \mathrm{TQ}(\Pi, g_0g_1g_2^2g_3) + \mathrm{TQ}(\Pi, g_0g_1g_2^2g_3^2) +$
      $\mathrm{TQ}(\Pi, g_0g_1^2g_2g_3) + \mathrm{TQ}(\Pi, g_0g_1^2g_2g_3^2) + \mathrm{TQ}(\Pi, g_0g_1^2g_2^2g_3) + \mathrm{TQ}(\Pi, g_0g_1^2g_2^2g_3^2) +$
      $\mathrm{TQ}(\Pi, g_0^2g_1g_2g_3) + \mathrm{TQ}(\Pi, g_0^2g_1g_2g_3^2) + \mathrm{TQ}(\Pi, g_0^2g_1g_2^2g_3) + \mathrm{TQ}(\Pi, g_0^2g_1g_2^2g_3^2) +$
      $\mathrm{TQ}(\Pi, g_0^2g_1^2g_2g_3) + \mathrm{TQ}(\Pi, g_0^2g_1^2g_2g_3^2) + \mathrm{TQ}(\Pi, g_0^2g_1^2g_2^2g_3) + \mathrm{TQ}(\Pi, g_0^2g_1^2g_2^2g_3^2) \neq 0$ then
    true
  else
    false
  endif.
```

Figure 1. Conflict detection algorithm for polynomial trajectories

Example 4. *Consider the following two polynomial trajectories, which appears as an example in.*[11]

$$\begin{aligned}
P_o(t) = (&-3.2484 + 270.7\,t + 433.12\,t^2 - 324.83999\,t^3,\\
&15.1592 + 108.28\,t + 121.2736\,t^2 - 649.67999\,t^3,\\
&38980.8 + 5414.0\,t - 21656.0\,t^2 + 32484.0\,t^3),\\
P_i(t) = (&1.0828 - 135.35\,t + 234.9676\,t^2 + 3248.4\,t^3,\\
&18.40759 - 230.6364\,t - 121.2736\,t^2 - 649.67999\,t^3,\\
&40280.15999 - 10828.0\,t + 24061.9816\,t^2 - 32484.0\,t^3).
\end{aligned}$$

The unit of time for these trajectories is hours (hr), the unit of horizontal position is nautical miles (nmi), and the unit of vertical position is feet (ft). At time $t = 0$ hours (current time), the positions of the ownship and intruder aircraft are $(-3.2484, 15.1592, 38980.8)$ and $(1.0828, 18.40759, 40280.15999)$, respectively. At this time, the aircraft are approximately 5.414 nmi apart horizontally and approximately 1299.36 ft apart vertically. Thus, given the separation standard minima of 5 nmi horizontally and 1000 ft vertically, the aircraft are not currently in loss of separation.

*The algorithm **cd_poly** predicts that the aircraft are in conflict for a lookahead time of 3 minutes, i.e., when $T = \frac{1}{20}$. That is **cd_poly**(P_o, P_i) = **true**. In fact, it is shown in*[11] *that the aircraft are in loss of separation at time $t = \frac{5105}{262144}$, or in about 70 seconds. It follows that conflict?(P_o, P_i) holds. At this time, the aircraft are approximately 4.999 nmi apart horizontally and -999.92 ft vertically.*

V. Related Work and Conclusion

Safety properties, including soundness, completeness and correctness, have been formally verified for CD&R algorithms that assume a linear trajectory model.[3,9,10,12] A conflict resolution algorithm for curved trajectories has been formally verified using hybrid-model checking techniques.[17] Other type of trajectories, such as piece-wise linear trajectories also enable analytic detection methods[6,7] and thus, formal proofs of these algorithms are feasible. CD&R algorithms that handle complicated nonlinear trajectories either iterate conflict computations at specified discrete steps[5,15] or they rely on approximation methods.[2,16,18] Formal verification of these kinds of algorithms is usually difficult. In,[11] the authors proposed a conflict detection algorithm for arbitrary trajectory models and they verified in PVS that the algorithm is correct within some approximation bounds. That is, the algorithm can be configured to be sound or complete, but not both.

This papers presents a conflict detection algorithm for two aircraft flying polynomial trajectories. The algorithm precisely determines whether the aircraft are in conflict within a given lookahead time. The proposed algorithm is sound and complete, i.e., it detects all conflicts and present no false alarms. To the best knowledge of the authors, this is the first conflict detection algorithm for nonlinear trajectory models that has been formally proved to be correct. While the algorithm presented in this paper assumed a trajectory model based on polynomial functions, this is not a significant limitation. Indeed, every nonlinear trajectory can be uniformly approximated with a polynomial trajectory in the time variable, for instance using Taylor series. This is because any continuous function can be uniformly approximated by polynomials.[19] In addition, there exist some models for turning trajectories, such as those based on splines, that are explicitly defined using polynomials.

The mathematical development presented in this paper, including definitions and theorems, has been specified and verified in the interactive theorem prover PVS. A theorem prover is a computer program that provides a specification language and a logic engine that checks every deduction step of a mathematical proof. This verification process is resource-intensive, but the safety critical role that CD&R systems play in the airspace system largely justifies this formalization effort.

References

[1] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] Antonio Bicchi and Lucia Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):221–231, December 2000.

[3] Gilles Dowek, Alfons Geser, and César Muñoz. Tactical conflict detection and resolution in a 3-D airspace. In *Proceedings of the 4th USA/Europe Air Traffic Management R&DSeminar, ATM 2001*, Santa Fe, New Mexico, 2001. A long version appears as report NASA/CR-2001-210853 ICASE Report No. 2001-7.

[4] Michael Eisermann. The fundamental theorem of algebra made effective: An elementary real-algebraic proof via Sturm chains. *The American Mathematical Monthly*, 119(9):715–752, November 2012.

[5] Heinz Erzberger and Karen Heere. Algorithm and operational concept for resolving short-range conflicts. In *Proceedings of the 26th International Congress of the Aeronautical Sciences, ICAS 2008*, Anchorage, Alaska, USA, September 2008.

[6] George Hagen, Ricky Butler, and Jeffrey Maddalon. Stratway: A modular approach to strategic conflict resolution. In *Preceedings of 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Virgina Beach, VA, September 2011.

[7] David A. Karr and Robert A. Vivona. Conflict detection using variable four-dimensional uncertainty bounds to control missed alerts. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, number AIAA 2006-6057, Keystone, Colorado, USA, August 2006.

[8] James Kuchar and Lee Yang. A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):179–189, December 2000.

[9] Jeffrey Maddalon, Ricky Butler, Alfons Geser, and César Muñoz. Formal verification of a conflict resolution and recovery algorithm. Technical Report NASA/TP-2004-213015, NASA Langley Research Center, NASA LaRC,Hampton VA 23681-2199, USA, April 2004.

[10] Jeffrey Maddalon, Ricky Butler, César Muñoz, and Gilles Dowek. Mathematical basis for the safety analysis of conflict prevention algorithms. Technical Memorandum NASA/TM-2009-215768, NASA, Langley Research Center, Hampton VA 23681-2199, USA, June 2009.

[11] Anthony Narkawicz and César Muñoz. Formal verification of conflict detection algorithms for arbitrary trajectories. *Reliable Computing*, 17:209–237, December 2012.

[12] Anthony Narkawicz, César Muñoz, and Gilles Dowek. Provably correct conflict prevention bands algorithms. *Science of Computer Programming*, 2011. In Press.

[13] Michael S. Nolan. *Fundamentals of Air Traffic Control*. Brooks Cole, 4 edition, July 2003.

[14] Sam Owre, John Rushby, and Natarajan Shankar. PVS: A prototype verification system. In Deepak Kapur, editor, *Proceeding of the 11th International Conference on Automated Deductioncade*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752. Springer, June 1992.

[15] Russell Paielli. Modeling manuever dynamics in air traffic conflict resolution. *Journal of Guidance, Control, and Dynamics*, 26(3):407–215, May–June 2003.

[16] Lucia Pallottino, Eric Feron, and Antonio Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE Transactions on Intelligent Transportation Systems*, 3(1):3–11, March 2002.

[17] André Platzer and Edmund M. Clarke. Formal verification of curved flight collision avoidance maneuvers: A case study. In Ana Cavalcanti and Dennis Dams, editors, *FM 2009: Formal Methods, 16th International Symposium on Formal Methods*, volume 5850 of *Lecture Notes in Computer Science*, pages 547–562. Springer, 2009.

[18] Arthur Richards and Jonathan How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of the 2002 American Control Conference*, volume 3, pages 1936–1941, 2002.

[19] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, third edition, 1976.

[20] Frank Sottile. Chapter 2: Real solutions to univariate polynomials. [Course Notes].

[21] Charles Sturm. Mémoire sur la résolution des équations numériques. In Jean-Claude Pont, editor, *Collected Works of Charles François Sturm*, pages 345–390. Birkhäuser Basel, 2009.

[22] Harry Swenson, Richard Barhydt, and Michael Landis. Next Generation Air Transportation System (NGATS) Air Traffic Management (ATM)-Airspace Project, June 2006.