Deterministic and Probabilistic Conflict Detection Algorithms for NextGen Airport Surface Operations

Sai Vaddi¹, Jason Kwan², Andrew Fong³, and Victor H. L. Cheng⁴
Optimal Synthesis Inc., Los Altos, CA, 94022

The paper deals with the development a ground-side conflict detection automation system for NextGen airport surface operations. The automation system is referred to as “Monitor Airport Environment: Surface Traffic and Runway Operations (MAESTRO).” In contrast to current-day conflict detection systems, MAESTRO has been designed taking into account NextGen operational concepts from mid-term and far-term timeframes. Conflicts of interest are Taxiway Collisions and Runway Incursions. A new conflict alert referred to as “Runway Incursion Situation Alert (RISA)” is created to actively prevent runway incursions. The automation system is driven by surveillance inputs and the outputs from airport planning systems such as Spot and Runway Departure Advisor (SARDA). MAESTRO consists of three modules: (i) Trajectory Prediction module, (ii) Conflict Detection module, and (iii) Controller Display module. The trajectory prediction module generates the 4D-trajectory predictions along with their uncertainty estimates. The paper develops the framework for both deterministic and probabilistic conflict detection. MEASTRO has been tested using actual surface traffic data from Dallas/Fort Worth International Airport (DFW). The evaluations indicate promising performance with zero missed-alerts and few false alarms that are actually close encounters. It is shown that situations which could potentially become Runway Incursions could be detected as RISAs with a lead-time of 60 seconds.

I. Introduction

Ensuring the safety of the National Airspace System (NAS) in the face of increasing traffic and congestion is of utmost significance. The recent dramatic incident between an A380 and CRJ-700 at the JFK airport¹ is a stark reminder of the safety issues affecting the NAS. The FAA’s NextGen Implementation Plan³ recognizes airport congestions as a major problem of the NAS. The plan includes airport expansion plans to build new runways, extend existing runways to accommodate larger aircraft with higher passenger capacities, relocate runways to increase lateral separation to allow parallel operations under Instrument Flight Rules (IFR), and build additional taxiways to accommodate the increased surface traffic. Successful implementation of these expansion plans means more complex airport layouts for the major airports, and more traffic operating on their surfaces. For airports with added runways, more flights need to cross active runways. Furthermore, new technologies that improve runway capacity through reduction in longitudinal separation will reduce the opportunity for active-runway crossing, compounding the runway-crossing problem. Major airports such as Dallas/Fort Worth International Airport (DFW) exemplify such complexity with as many as 7 runways. The NextGen concept²³ proposes the use of ground-based automation to schedule surface traffic and generate 4D taxi clearances to enable precise departure times and limited simultaneous runway occupancy. 4D Trajectory-based-operations could use tighter separations to improve the efficiency which would increase the potential for conflicts. Therefore, conflict detection capability becomes critical.

¹ Senior Research Scientist, 95 First Street, AIAA Member.
² Research Engineer, 95 First Street.
³ Research Engineer, 95 First Street.
⁴ Principal Scientist, 95 First Street, AIAA Associate Fellow.
The following sub-sections provide a literature survey of the past research on conflict detection systems both on the ground-side and flight-deck-side. Section II presents the functional architecture of MAESTRO. Section III describes the inputs to MAESTRO. Section IV briefly explains the Trajectory Prediction module of MAESTRO. Section V lists the conflict definitions used in this research. Section VI and Section VII present the deterministic and probabilistic conflict detection frameworks respectively. Section VIII presents the results obtained by testing MAESTRO with actual DFW surface traffic.

A. Past Research on Ground-Side CD&R

Current-day operations require the Air Navigation Service Provider (ANSP) to specify the taxi routes, control the order of merging at intersections, sequence runway crossings and departures at the runways, and require the pilots to provide separation visually. To enhance situational awareness of the ANSP, the FAA is introducing new surface surveillance technologies such as Airport Surface Detection Equipment – Model X (ASDE-X)^7 and Automatic Dependent Surveillance – Broadcast (ADS-B)^5, which provide aircraft position data in all-weather situations and support the prediction of future aircraft trajectories more accurately than before. Other technologies useful for conflict and incursion detection or prevention include the Airport Movement Area Safety System (AMASS)^6 and Runway Status Lights^8. The Runway Incursion Alerting System (RIAS)^9 consisting of millimeter-wave radar and pan/tilt/zoom cameras was developed by QinetiQ. The EUROCONTROL Advanced Surface Movement Guidance and Control System (A-SMGCS)^10 concept includes research on optimization of airport taxi scheduling^11. A-SMGCS Level 2 consists of automated monitoring and alerting functions, and includes the prediction of conflicts on active runways or incursions into restricted areas. The European Airport Movement Management by A-SMGCS (EMMA) project defined A-SMGCS operational requirements^12 for the ANSP and flight deck, and other important services such as Communications, Navigation, and Surveillance (CNS)^13. Further development of A-SMGCS services, procedures, and operational requirements has been documented as part of the EMMA2 effort^14.

B. Past Research on Flight-Deck-Side CD&R

A detailed literature survey of the flight deck CD&R systems is provided in Ref. 15. Some of these systems are: (i) Runway Awareness and Advisory System (RAAS) developed by Honeywell International Inc, and (ii) SafeRoute™ developed by Aviation Communication & Surveillance Systems (ACSS), and (iii) PathProx™ developed by Era Corporation in collaboration with NASA. Previous NASA research for improving situational awareness on the flight deck include the Taxiway Navigation and Situation Awareness (T-NASA) System^16,17 developed at NASA Ames, and the Runway Incursion Prevention System (RIPS)^18,19 developed at NASA Langley. Researchers at NASA Langley are also building on the earlier RIPS technologies to develop flight-deck technologies for collision avoidance^20. NASA is conducting Collision Avoidance for Airport Traffic (CAAT) research^21–23 to develop technologies, data, and guidelines to enable Conflict Detection and Resolution (CD&R) in the Airport Terminal Maneuvering Area (ATMA) under current and emerging NextGen operating concepts. The research led to the development of a flight-deck CD&R tool referred to as Airport Traffic Collision Avoidance Monitor (ATCAM).

II. MAESTRO

Most of the conflict detection system presented in the previous section are based on surveillance data. They are tactical in nature and are based on dead-reckoning projections. The current work described in this paper deals with algorithms for conflict detection with emphasis on mid-term and far-term trajectory-based surface operations. Detailed concept of requirements for MAESTRO have been presented in Refs. 24–26. It is assumed that planners such as NASA’s Spot and Runway Departure Advisor (SARDA) or Optimal Synthesis Inc.’s GoSAFE (Ground-Operation Situation Awareness and Flow Efficiency) concept^27,28 will provide intent information for predicting conflicts. Surface Operation Automation Research (SOAR) forms the seminal research in surface 4DT operations in a holistic approach to the problem. Researchers from Optimal Synthesis Inc. (OSI) have developed conflict-free surface operations planners as part of the SOAR concept. OSI’s surface operation planner is based on the GoSAFE concept. GoSAFE handles the taxiway route assignment, runway assignment, taxiway sequencing and scheduling, departure runway scheduling, runway exit assignment and scheduling, and runway crossing operations. The SARDA concept and implementation of optimized airport surface traffic operations has been presented by SESO researchers in Ref. 29. The concept consists of a spot release planner^30 and a runway scheduler^31,32. Even though SARDA uses a trajectory-based design of schedule, it does not issue trajectories as clearances. MAESTRO is developed to take advantage of additional intent information resulting from the airport operational planners such as SARDA. The SARDA planners generates information such as: (i) taxiway routes starting from ramp spot till the
runway, (ii) ramp spot release time, and (iii) departure runway sequence. Other planners such as GoSAFE generate 4D-routes consists of (i) taxiway routes, and (ii) Required Times of Arrival (RTAs) for some or all intermediate nodes. The additional intent information results in better trajectory prediction and hence better conflict detection. Figure 1 shows the overall functional block diagram of MAESTRO. The outputs from MAESTRO consist of Conflict Alerts that could be displayed on the Tower Automation display and optionally datalinked to the flight-deck for flight crew display. A separate paper Ref. 33 by the same authors evaluates various ground-air integration options.

![Figure 1. Functional Block Diagram of MAESTRO](image)

The following section describes the external inputs to MAESTRO and the pre-processing of those inputs by MAESTRO.

### III. Inputs to MAESTRO

#### A. External Inputs

The following are the external inputs to MEASTRO:

- Surveillance Data (The concept and requirements document for MAESTRO\textsuperscript{24} discusses the different surveillance options and error characteristics.)
  - List of aircraft detected by the surveillance system and identified by their ids: 
    \[
    \{\text{acid}_1, \text{acid}_2, \text{acid}_3, \ldots, \text{acid}_n\}
    \]
  - The set of current positions of these aircraft:
    \[
    \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)\}
    \]
    where \((x_i, y_i, z_i)\) are the position coordinates as detected by the surveillance system.
  - The set of current speed, heading, and flight path angle of these aircraft:
    \[
    \{(V_1, \chi_1, \gamma_1), (V_2, \chi_2, \gamma_2), \ldots, (V_n, \chi_n, \gamma_n)\}
    \]
where \( V_i, Z_i, \gamma_i \) are the magnitude of ground-speed vector, heading angle of the ground-speed vector, and flight path angle, respectively, for the \( i^{th} \) aircraft.

- **Planner Data**
  - The set of planned routes for each aircraft \( \{ R_1, R_2, \cdots, R_n \} \). Each route consists of a sequence of nodes and links.
  - The scheduled times of release from ramp-spot for departures \( \{ t_{rs1}, t_{rs2}, \cdots, t_{rsn} \} \).
  - The planned sequence of departures

**B. Pre-Processing of Inputs**

MAESTRO uses a geometric link-node model of the airport to map the positions of the aircraft to a link and a domain. The augmented positional states of the aircraft can be written as follows:

\[
\begin{aligned}
(x_1, y_1, z_1, l_1, D_1), (x_2, y_2, z_2, l_2, D_2), \cdots, (x_n, y_n, z_n, l_n, D_n)
\end{aligned}
\]

where \( l_i \) is the link, and \( D_i \) is the airport model domain of the \( i^{th} \) aircraft. Candidate aircraft domains are as follows: (i) Ramp Areas, (ii) Runway Approaches, (iii) Runways, (iv) Runway Hold Pads, (v) Runway Exits, (vi) Runway Crossings, and (vii) Taxiways. The domains are modeled as convex polygons as seen in Figure 2-4. The polygon representation results in a mathematical representation as a system of linear inequalities that can be easily evaluated to determine whether the aircraft location falls within a polygon. Each polygon is represented by an inequality that looks as shown below:

\[
F \begin{bmatrix} x \\ y \end{bmatrix} + G \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{1}
\]

where \( F \) is a 4x2 matrix and \( G \) is a 4x1 vector.

Figure 2. Runway Polygons (red), Hold Pads (blue)  
Figure 3. Runway Crossing Polygons (red)
The following section describes MAESTRO’s Trajectory Prediction module which is the pre-cursor to conflict detection.

**IV. Trajectory Prediction**

Trajectory prediction is a crucial component of MAESTRO. It is the primary input that drives the conflict detection module. MAESTRO’s Trajectory Prediction (TP) module is based on point-mass acceleration level aircraft equations of motion. TP models the following different phases of an arrival aircraft: (i) final approach, (ii) flare, (iii) landing roll, (iv) runway exit, and (iv) taxiway motion. For departure aircraft the TP module models the following phases: (i) taxiway motion, (ii) takeoff roll, and (iii) takeoff. MAESTRO uses aircraft performance models to model the aircraft’s speed, acceleration, altitude rate, and turn rate in each of these modes. Departure aircraft paths are segmented as follows:

Gate → Ramp Spot → Taxiway Nodes → Runway Crossing Node → Departure Runway Entry Node

Arrival aircraft paths are segmented as follows:

Outer Marker → Runway Threshold → Runway Exit Node → Taxiway Nodes
→ Runway Crossing → Ramp Spot → Gate

Table 1 and Table 2 describe the methodology for selecting the (i) initial position, (ii) initial time, (iii) final position, and (iv) final time of the trajectory predictor for departures and arrivals, respectively. For the sake of brevity, complete details of the TP module are not included in this paper.
### Table 1. Departure Aircraft

<table>
<thead>
<tr>
<th>Current Aircraft Path Segment</th>
<th>Initial Position &amp; Initial Time for Trajectory Prediction</th>
<th>Final Position &amp; Final Time for Trajectory Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft is in the ramp area and has not reached the ramp spot.</td>
<td>Initial position: Ramp spot assigned by the SARDA planner. Initial time: Ramp spot release time specified by the SARDA planner.</td>
<td>Position achieved by propagating the aircraft over the chosen trajectory prediction time horizon along taxiway route or a runway crossing node or a departure runway queue node, whichever comes first. It is expected that the aircraft will stop at the runway crossing or at the departure runway.</td>
</tr>
<tr>
<td>Aircraft has left the ramp spot and is on the taxiway.</td>
<td>Initial Position: Current surveillance position Initial Time: Current time</td>
<td>Same as above</td>
</tr>
<tr>
<td>Aircraft is stationary at the runway crossing.</td>
<td>Same as above</td>
<td>Aircraft is assumed to stay stationary until surveillance data indicates movement.</td>
</tr>
<tr>
<td>Aircraft has started to cross the runway.</td>
<td>Same as above</td>
<td>Position achieved by propagating the aircraft over the chosen trajectory prediction time horizon or another runway crossing node or a departure runway queue node, whichever comes first. It is expected that the aircraft will stop at the runway crossing or at the departure runway.</td>
</tr>
<tr>
<td>Aircraft is waiting in the departure runway queue.</td>
<td>Same as above</td>
<td>Aircraft is assumed to stay stationary until surveillance data indicates movement.</td>
</tr>
<tr>
<td>Aircraft is on runway has started to takeoff.</td>
<td>Same as above</td>
<td>Position achieved by propagating the aircraft over the chosen time horizon or until the aircraft reaches a certain altitude that restricts the scope of the ground-side CD&amp;R.</td>
</tr>
</tbody>
</table>

### Table 2. Arrival Aircraft

<table>
<thead>
<tr>
<th>Current Aircraft Path Segment</th>
<th>Initial Position &amp; Time for Trajectory Prediction</th>
<th>Final Position &amp; Time for Trajectory Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft is in the terminal area and has not reached the outer marker.</td>
<td>Initial position: Outer marker Initial time: Predicted time of arrival at the outer marker. If the predicted time of arrival is outside the prediction time horizon then no trajectory prediction is made.</td>
<td>Position achieved by propagating the aircraft over the chosen time horizon or the runway exit node, whichever comes first.</td>
</tr>
<tr>
<td>Aircraft is on the glide slope, or is in the flare maneuver, or is on the runway.</td>
<td>Initial Position: Current surveillance position Initial Time: Current time</td>
<td>Same as above</td>
</tr>
<tr>
<td>Aircraft is waiting at the runway crossing or a runway exit node.</td>
<td>Same as above</td>
<td>Aircraft is assumed to stay stationary until surveillance data indicates movement.</td>
</tr>
<tr>
<td>Aircraft has started to cross the runway.</td>
<td>Same as above</td>
<td>Position achieved by propagating the aircraft over the chosen time horizon or the next runway crossing node or the assigned ramp spot, whichever comes first.</td>
</tr>
</tbody>
</table>
Aircraft is on the taxiway.  

Same as above  

Position achieved by propagating the aircraft over the chosen time horizon or the assigned ramp spot, whichever comes first.

Aircraft are propagated using performance characteristics of individual aircraft type such as: (i) nominal taxi speed, (ii) nominal turning speed, (iii) nominal taxiway acceleration levels, (iv) nominal taxiway deceleration levels, (v) takeoff speeds, (vi) landing speeds, (vii) landing roll deceleration, and (viii) flare duration.

The set of trajectory predictions can be written as \( \{T_1, T_2, \ldots, T_n\} \) where, \( T_i \) is the trajectory prediction for the \( i^{th} \) aircraft, which can be further expanded as follows:

\[
T_i = \{(t_{i0}, s_{i0}, x_{i0}, y_{i0}, z_{i0}),(t_{i1}, s_{i1}, x_{i1}, y_{i1}, z_{i1}),\ldots\}
\]  

where, \( x_{ij}, y_{ij}, \text{ and } z_{ij} \) are the \( x, y, \text{ and } z \) coordinate predictions for the \( i^{th} \) aircraft at the prediction time instance \( t_{ij} \), \( s_{ij} \) is the path length variable which is defined by the horizontal plane coordinates \( x_{ij}, y_{ij} \) and the Route \( R_i \). It should be noted that the path length variable is a scalar representation of the two horizontal plane coordinates. The time step for prediction time instance need not be equal but they should be fine enough for the purposes of conflict detection.

Figure 5 shows the path length errors \( \Delta s_{ij} = s_{\text{actual},ij} - s_{\text{predicted},ij} \) resulting from trajectory prediction when tested with actual DFW surface trajectory data for MD82 aircraft type. Figure 6 shows the distribution of errors for 60s trajectory predictions.

The stochastic nature of the errors as shown in Figure 5-Figure 6 motivate the need for a probabilistic representation of the trajectory predictions, which can be written as follows:

\[
T_i = \{(t_{i0}, P_{s\rightarrow 0}(s_{i0}), P_{z\rightarrow 0}(z_{i0})),(t_{i1}, P_{s\rightarrow 1}(s_{i1}), P_{z\rightarrow 1}(z_{i1})),\ldots\}
\]  

where, \( s_{ij} \) is the predicted path length value, and \( z_{ij} \) is the predicted altitude of the \( i^{th} \) aircraft at the \( j^{th} \) prediction time instance \( t_{ij} \). The path length variable is used as opposed to the position coordinates because as it is better suited to model the uncertainty in predictions. In this prediction mode, aircraft are expected to stay along their planned route. Therefore, the two dimensional path in the horizontal plane can be described using a single parameter, which is chosen as the path length. Predictions of the path length variable \( s_{ij} \) and the altitude \( z_{ij} \) of the \( i^{th} \) aircraft are treated as random variables that are described using their probability distribution functions.

\( P_{s\rightarrow ij} \) is the probability density function associated with the path length variable for the \( i^{th} \) aircraft at the prediction time instance \( t_{ij} \).
$p_{z_{ij}}$ is the probability density function associated with the altitude variable for the $i^{th}$ aircraft at the prediction time instance $t_{ij}$.

The Gaussian probability distribution requires two parameters for complete description, the mean and the variance.

$$T_i = \{t_{i0}, s_{10 \mu}, s_{10 \sigma}, t_{i1}, s_{11 \mu}, s_{11 \sigma}, t_{i2}, s_{12 \mu}, s_{12 \sigma}\}$$

(4)

where, $s_{ij \mu}$ and $s_{ij \sigma}$ are the mean and standard deviation of the path length variable prediction for the $i^{th}$ aircraft at the prediction time instance $t_{ij}$. Similarly, $z_{ij \mu}$ and $z_{ij \sigma}$ are the mean and standard deviation of the altitude prediction for the $i^{th}$ aircraft at the prediction time instance $t_{ij}$.

A generic continuous probability distribution could require infinite parameters which is not suitable for implementation as part of the conflict detection algorithms. A discretized finite length representation of the continuous probability distribution function is ideal for the purpose of the conflict detection. It can be used for any probability distribution including common ones such as ‘Gaussian’ and ‘Uniform’ probability distribution functions. Generic discretized representations of the continuous probability distribution function are given below:

$$p_{s_{ij}} = \left[\left(s_{ij \min}, p_{s_{ij \min}}\right), \left(s_{ij \mu 1}, p_{s_{ij \mu 1}}\right), \ldots, \left(s_{ij \mu m}, p_{s_{ij \mu m}}\right)\right]$$

(5)

$$p_{z_{ij}} = \left[\left(z_{ij \min}, p_{z_{ij \min}}\right), \left(z_{ij \mu 1}, p_{z_{ij \mu 1}}\right), \ldots, \left(z_{ij \mu m}, p_{z_{ij \mu m}}\right)\right]$$

(6)

where,

$$p_{s_{ij \mu}} = \text{Prob}(s_{ij \mu} \leq s_{ij \mu 1}) = \int_{-\infty}^{s_{ij \mu 1}} p_{s_{ij \mu}} ds$$

(7)

$$p_{z_{ij \mu}} = \text{Prob}(z_{ij \mu} \leq z_{ij \mu 1}) = \int_{-\infty}^{z_{ij \mu 1}} p_{z_{ij \mu}} dz$$

(8)

The values $s_{ij \min}$ and $s_{ij \max}$ are to be chosen such that $p_{s_{ij \min}}$ and $(1 - p_{s_{ij \max}})$ are very small quantities. In other words the probability that the path length variables assumes a value smaller than $s_{ij \min}$ or greater than $s_{ij \max}$ should be negligible.

The following section lists the conflict definitions adopted in the current research.

V. Conflict Definitions

Two types of conflicts on the airport surface are considered under the current research: (i) taxiway collisions, and (ii) runway incursions. These conflicts are further described in the following sub-sections.

A. Collision Conflict Definition

Criteria for collision conflicts:

$$z_1 = z_2 = 0 \text{ and } \Delta xy_{12} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} < \Delta xy_{tol}$$

(9)

The separation requirement $\Delta xy_{tol}$ is dependent on the dimensions and the orientations of the two aircraft. The separation requirement is illustrated for different collision scenarios in Figure 7 to Figure 10. Aircraft length and wingspan are represented using the symbols ‘L’ and ‘W’ in these figures.
The function $C_{\text{collision}}$ for detecting collision conflicts between a pair of aircraft takes in two arguments, one for each aircraft. The argument for each aircraft in turn consists of six pieces of data: $(x, y, l, \chi, L, W)$, representing the horizontal-plane position coordinates, link associated with the aircraft, heading angle of the aircraft, length of the aircraft, and the wingspan of the aircraft. The primary output of the function takes two values 0 or 1 depending on whether the two aircraft satisfy the detection criteria for collision or not.

$$C_{\text{collision}} = \begin{cases} 1, & \text{if} \quad \Delta \chi_{12} < \Delta \chi_{12, \text{tol}} \quad \text{(10)} \\ 0, & \text{otherwise} \end{cases}$$

where $\Delta \chi_{12, \text{tol}} = \max \left( \frac{L_1}{2} + \frac{L_2}{2}, \frac{W_1}{2} + \frac{W_2}{2}, \frac{L_1}{2} + \frac{L_2}{2}, \frac{W_1}{2} + \frac{W_2}{2}, \frac{L_1}{2} + \frac{L_2}{2} \right)$ For head-on collisions the detection logic uses an additional check. If two aircraft occupy the same link $l_1 = l_2$ at the same time and are on reciprocal heading $|\chi_1 - \chi_2| = 180^\circ$, then the aircraft are imminently headed for a head-on collision.

**B. Runway Incursion Definitions**

FAA defines runway incursion as any occurrence at an aerodrome involving the incorrect presence of an aircraft, vehicle, or person on the protected area of a surface designated for the landing and take-off of aircraft. An incursion may occur either because an aircraft or ground vehicle did not have the proper clearance, or because the controller made an error when issuing a clearance. It is difficult to evaluate the FAA’s definition of runway incursion without information related to voice-based clearances. The following paragraphs describe the runway incursion definition adopted in this research. Figure 11-Figure 14 illustrate a few runway incursion scenarios.

Runway Incursion Scenario 1 (Figure 11):

- Aircraft A is on Runway 1 and moving; Aircraft B is also on Runway 1 in front of A
Runway Incursion Scenario 2 (Figure 12):
- Aircraft A on Runway 1 and moving; Aircraft B is on a crossing of Runway 1 Crossing in front of A

Runway Incursion Scenario 3 (Figure 13):
- Arrival Aircraft A is on the threshold of Runway 1 about to land; Aircraft B is on the crossing of Runway 1 front of A

Runway Incursion Scenario 4 (Figure 14):
- Aircraft A is on Runway 1 before the intersection and moving; Aircraft B is on a Intersecting Runway 2 also before the intersection and moving
Figure 14. Runway Incursion Involving Intersecting Runways (Scenario 4)

The function $C_{\text{runway,incursion}}$ detection runway incursions between a pair of aircraft. The output of the function takes two values 0 or 1 depending on whether the two aircraft satisfy the detection criteria for runway incursion or not.

$$C_{\text{runway,incursion}}(D_1, l_1, n_1, z_1, V_1, l_2, n_2, z_2, V_2) \in [0,1]$$ (12)

The function $C_{\text{runway,incursion}}$ treats the following scenarios as runway incursions if Aircraft 1 is on the runway approach and below the Missed Approach Point (MAP), i.e., $D_1 = \text{Runway Approach}$ and $z_1 < MAP_{\text{altitude}}$:

- $(D_2 = \text{Runway}) \land (h_2 \leq h_{\text{to, tol}})$
- $D_2 = \text{Runway Crossing}_j$
- $(D_2 = \text{Runway Approach}) \land (z_2 < MAP_{\text{altitude}}) \land (\text{Nodes}_{\text{rwy}, i} \cap \text{Nodes}_{\text{rwy}, j} \neq \emptyset)$
- $(D_2 = \text{Runway}) \land (V_2 > 0) \land (\text{Nodes}_{\text{rwy}, i} \cap \text{Nodes}_{\text{go, rwy}, j} \neq \emptyset)$

where $MAP_{\text{altitude}}$ represents the altitude of the missed approach point. The variable $h_{\text{to, tol}}$ is a altitude tolerance to determine if the aircraft is airborne. The set $\text{Nodes}_{\text{go, rwy}, j}$ is computed as the set of nodes from $\text{Nodes}_{\text{rwy}, j}$ that are ahead of the current node $n_2$. It should be noted that the runway nodes are sequenced in the same order as the direction of the runway. The set $\text{Nodes}_{\text{go, rwy}, j}$ includes all nodes till the end of the runway even if the planned aircraft route deviates from the runway at an intermediate exit.

The function $C_{\text{runway,incursion}}$ also treats the following scenarios as runway incursions if Aircraft 1 is on the runway and moving, i.e., $D_1 = \text{Runway}_{\text{go}}$, and $V_1 > V_{\text{lim}}$:

- $(D_2 = \text{Runway}) \land (l_2 \in \text{Links}_{\text{go, rwy}, i}) \land (h_2 \leq h_{\text{to, tol}})$
- $(D_2 = \text{Runway}) \land (\text{Links}_{\text{go, rwy}, i} \cap \{l_2\} \neq \emptyset) \land (h_2 \leq h_{\text{to, tol}})$
- $(D_2 = \text{Runway Crossing}_g) \land (\text{Nodes}_{\text{go, rwy}, i} \cap \text{Nodes}_{\text{crossing, g}} \neq \emptyset)$
- $(D_2 = \text{Runway}) \land (z_2 < MAP_{\text{altitude}}) \land (\text{Nodes}_{\text{go, rwy}, i} \cap \text{Nodes}_{\text{rwy}, j} \neq \emptyset)$
- $(D_2 = \text{Runway}) \land (V_2 > 0) \land (\text{Nodes}_{\text{go, rwy}, i} \cap \text{Nodes}_{\text{go, rwy}, j} \neq \emptyset)$

where the set $\text{Links}_{\text{go, rwy}, i}$ is computed as all links from $\text{Links}_{\text{rwy}, i}$ that are ahead of the current link $l_1$. The set $\text{Nodes}_{\text{go, rwy}, i}$ is computed as the set of nodes from $\text{Nodes}_{\text{rwy}, i}$ that are ahead of the current node $n_1$. The set
Nodes_{go\_rwy\_j} is computed as the set of nodes from \( Nodes_{rwy\_j} \) that are ahead of the current node \( n_2 \). The variable \( V_{lim} \) is a threshold speed chosen to determine the intent to takeoff.

C. Runway Incursion Situation Alert (RISA) Definition

Under current day operations Runway Incursions involving aircraft on ground can only be reliably predicted with lead times up to 10 seconds. This is due to the fact that taxiing aircraft are capable of coming to a halt in 10 seconds. Runway Incursion Situation Alert is created to address this deficiency. The following are the features of RISA:

- RISAs identify situations where an “Aircraft is Expected to Be in the Vicinity of an Active Runway”
- RISAs are not Runway Incursions, but very close to result in one
- RISAs can be generated with a lead time up to 60 seconds
- RISAs can be communicated to Flight-Deck over Datalink
- RISAs can prevent runway incursions
- RISAs are equivalent to a predictive “Runway Status Light” with the display on the flight-deck-side.

Figure 12 shows a Runway Incursion scenario involving a departure aircraft and a crossing aircraft, where the crossing aircraft enters the runway crossing at the same time the departure aircraft has initiated takeoff roll. Figure 15 shows a similar scenario that involves the departure aircraft waiting to takeoff and the crossing aircraft waiting to cross the runway. This scenario in Figure 15 does not constitute a Runway Incursion but is one mistake away from either pilot or the local controller to result in a Runway Incursion.

Similarly, Figure 13 shows a Runway Incursion scenario involving an arrival aircraft and a crossing aircraft, where the crossing aircraft enters the runway crossing at the same time the arrival aircraft has crossed the runway threshold. Figure 16 shows a similar scenario that involves the arrival aircraft on the final approach close to the runway threshold and the crossing aircraft waiting to cross the runway. Again, the scenario in Figure 16 does not constitute a Runway Incursion but is one mistake away from either the crossing pilot or the local controller to result in a Runway Incursion.

MAESTRO predicts and characterizes situations in Figure 15 and Figure 16 as RISAs. The RISAs can be transmitted over the datalink to flight-deck to actively and in a predictive manner indicate to the pilots the status of the runway. Thus, RISAs can prevent pilots from entering active runways and prevent Runway Incursions.

![Figure 15. RISA Involving Departure Aircraft and Crossing Aircraft](image-url)
The function \( C_{\text{risa}} \) detects situations that are close to a runway incursion but do not exactly satisfy the definition of runway incursion as RISAs:

\[
C_{\text{risa}} \left( [D_1 \ l_1 \ n_1 \ z_1 \ V_1], [D_2 \ l_2 \ n_2 \ z_2 \ V_2] \right) \in \{0,1\}
\]  

The function \( C_{\text{risa}} \) treats the following scenarios as RISA if Aircraft 1 is on the runway approach and below the Missed Approach Point (MAP), i.e., \( D_1 = \text{Runway Approach}_i \) and \( z_1 < \text{MAP}_{\text{altitude}} \):

- \( (D_2 = \text{Runway Vicinity}_i) \land \left( \text{Rwy Links}_i \cap \text{Links}_{\text{2go},2} \neq \phi \right) \lor \left( \text{Links}_i \cap \text{Links}_{\text{2go},2} \neq \phi \right) \)

where the set \( \text{Rwy Links}_i \) are all the links of \( i^{th} \) runway; \( \text{Links}_i \) is the planned route for Aircraft 1; \( \text{Links}_{\text{2go},2} \) is the route to go for the Aircraft 2 including its current link (all links of Aircraft 2 ahead of current link \( l_2 \)).

The function \( C_{\text{risa}} \) also treats the following scenarios as RISA if Aircraft 1 is on the runway, i.e., \( D_1 = \text{Runway} \):

- \( (D_2 = \text{Runway Vicinity}_i) \land \left( \text{Links}_{\text{2go,wy},i} \cap \text{Links}_{\text{2go},2} \neq \phi \right) \)

where the set \( \text{Links}_{\text{2go,wy},i} \) is computed as all links from \( \text{Links}_{\text{wy},i} \) that are ahead of the current link \( l_1 \) and \( \text{Links}_{\text{2go},2} \) is the route to go for the Aircraft 2 including its current link (all links of Aircraft 2 ahead of current link \( l_2 \)).

The following section describes the Deterministic Conflict Detection procedure. The procedure consists of evaluating the trajectory predictions using the conflict definitions presented in this section.

VI. Deterministic Conflict Detection

Deterministic conflict detection refers to conflict detection that is done on the basis of deterministic trajectory predictions. Deterministic conflict detection procedure adopts a crisp classification (\( P_{\text{conf}} \in [0,1] \)) of a pair of aircraft states as being in conflict or not. The conflict detection algorithm scans through the trajectory predictions of aircraft pairs for conflicts. It should be noted that the starting time and ending time for each trajectory prediction may not be the same. Conflicts between the aircraft are then evaluated only over the common prediction time period. Comparison of deterministic 4D trajectories for conflicts involves comparison of the individual aircraft states at each prediction time instant. Conflict parameters such as time to conflict, minimum separation, and duration of runway incursion are then evaluated by aggregation over the prediction time horizon. The following conflict parameters are computed by the deterministic conflict detection algorithm:

- **Time of initial conflict**: The first time instant of conflict detection. The conflict could be either a collision conflict or a runway incursion. It should be noted that the aircraft need not be in a state of collision at the time of conflict. This is especially true for runway incursions and head-on collision conflicts that are detected before the actual collision occurs.

- **Duration of conflict**: This is the contiguous duration over which a conflict lasts, starting with the time to conflict. This is more important for runway incursions, some of which can last a very few seconds. For example, consider the scenario where an arrival aircraft is very close to its exit and another aircraft crosses the runway in front of it. This meets the definition of runway incursion, but as soon as the aircraft exits the
runway it no longer meets the definition. In such cases evaluating the collision criteria over the duration of conflict helps determine the nature of conflict resolution to be adopted.

- **Time of collision**: The first time instant when $\Delta y_{ij} < \Delta y_{ij, tol}$. Time to collision is different from time to conflict for head-on collision conflicts and runway incursions. The time to conflict for a head-on collision is the first time instant when both the aircraft share the same link with reciprocal heading. However, the actual collision only occurs later. Similarly, the time of occurrence of a runway incursion and time at which a collision happens due to the runway incursion are different. The time of occurrence of runway incursion in this case would be the time of conflict, and the time at which the collision is predicted to occur is the time to collision.

- **Location of collision**: The point at which collision is expected to occur.

- **Minimum separation**: The minimum horizontal-plane separation between the two aircraft over the prediction time horizon. This is meaningful for evaluating the severity of runway incursion conflicts.

- **Time of minimum separation**: The time at which the minimum horizontal-plane separation occurs between the two aircraft.

### VII. Probabilistic Conflict Detection

Probabilistic conflict detection refers to conflict detection that is done on the basis of probabilistic trajectory predictions. As a result, probabilistic conflict detection results in probabilistic description of conflict. Instead of a crisp classification of conflicts, the conflicts are now characterized by their chance of occurrence. Probabilistic conflict detection is more suitable for long-term conflict detection. Probabilistic conflict detection also involves a different approach for processing the trajectory predictions. When using deterministic trajectory predictions, the states of the aircraft pairs at the same prediction time instant are compared. The process for probabilistic 4D-trajectory predictions is much more complicated. Whereas deterministic trajectory predictions are associated with one state value at each prediction time instant, probabilistic trajectory predictions are associated with infinite possibility of the state value as a probability distribution. Therefore, when dealing with probabilistic trajectory predictions, the following extra steps are needed: (i) identification of the probable path length intervals and the associated cumulative probability distribution function, and (ii) mapping the path length value to the horizontal-plane position coordinates of the aircraft. The above two steps are required to be done for all candidate conflicting aircraft. Two additional steps are required for comparing a pair of aircraft. The first step involves identification of the conflict-prone path length intervals of individual aircraft. The second step involves computation of the probability of conflict. The following sub-sections contain more details of each of these steps.

#### A. Probable Path Length Interval

For a normally distributed path length variable with mean $s_\mu$ and standard deviation $s_\sigma$, the probable path length interval can be chosen as:

$$ s_{prob} = [s_{min}, s_{max}] = [s_\mu - 3s_\sigma, s_\mu + 3s_\sigma] $$

Sample discretized cumulative probability distribution for a Gaussian distribution is listed in Table 3. In this case the path length variable is discretized in units of the standard deviation $s_\lambda = s_\sigma$.

<table>
<thead>
<tr>
<th>$s$</th>
<th>$P_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{min}$</td>
<td>0.13%</td>
</tr>
<tr>
<td>$(s_\mu - 2s_\sigma)$</td>
<td>2.28%</td>
</tr>
<tr>
<td>$(s_\mu - s_\sigma)$</td>
<td>15.87%</td>
</tr>
<tr>
<td>$s_\mu$</td>
<td>50%</td>
</tr>
<tr>
<td>$(s_\mu + s_\sigma)$</td>
<td>85.1345%</td>
</tr>
<tr>
<td>$(s_\mu + 2s_\sigma)$</td>
<td>97.25%</td>
</tr>
<tr>
<td>$s_{max} = (s_\mu + 3s_\sigma)$</td>
<td>99.865%</td>
</tr>
</tbody>
</table>

Table 3. Discretized Representation of a Gaussian Probability Distribution Function

American Institute of Aeronautics and Astronautics
 Whereas the Normal distribution and the Uniform distribution can be specified by two parameters, a generic probability distribution can be specified using the discretized cumulative probability distribution.

\[
\left[ \left( s_{min}, P_{s_{min}} \right), \left( s_{min} + s_{\Delta}, P_{s_{1}} \right), \left( s_{min} + 2s_{\Delta}, P_{s_{2}} \right), \ldots, \left( s_{max}, P_{s_{max}} \right) \right]
\]

where \( P_{s_{k}} = \text{Prob}(s \leq s_{min} + ks_{\Delta}) \) and the probable path length interval is simply \( s_{prob} = [s_{min}, s_{max}] \).

### B. Mapping Path Length Values to Position Coordinates

Probabilistic 4D-trajectory predictions require a mapping from the path length variable \( s \) to the \((x, y)\) position coordinates. The path length variable can be mapped to the horizontal-plane position coordinates using the route information.

\[
x_{ij-p} = \text{interp}(s_{ij-p}, s_{ij-r}, x_{ij-r})
\]

\[
y_{ij-p} = \text{interp}(s_{ij-p}, s_{ij-r}, y_{ij-r})
\]

It is assumed that the route is parameterized in terms of path length as follows:\( (s_{ij-r}, x_{ij-r}, y_{ij-r}) \).

Once the path length prediction is mapped to the position coordinates, the next step involves identifying the link and domain associated with the predicted \((x, y)\) position coordinates.

\[
l_{ij-p} = \text{Link}(x_{ij-p}, y_{ij-p}, z_{ij-p})
\]

\[
D_{ij-p} = \text{Domain}(x_{ij-p}, y_{ij-p}, z_{ij-p})
\]

### C. Probability-of-Conflict Computation

The probability of conflict between two aircraft can be written in terms of the probability distribution functions of their path length variables as shown in the following expression:

\[
P_{ij-conflict} = \frac{\infty}{0} \int C_{ij}(s_{i}, s_{j})p_{i}(s_{i})p_{j}(s_{j})ds_{i}ds_{j}
\]

where, \( p_{i}(s_{i}) \) and \( p_{j}(s_{j}) \) are the probability distribution functions of the path length variables of Aircraft \( i \) and Aircraft \( j \), respectively. \( C_{ij}(s_{i}, s_{j}) \) is a conflict indicator function that assumes the values of 0 or 1 depending on whether the two aircraft are in a state of conflict or not, for a given pair of path length variable values.

\[
C_{ij}(s_{i}, s_{j}) = \begin{cases} 
1, & \text{if } s_{i}, s_{j} \text{ are in conflict} \\
0, & \text{if } s_{i}, s_{j} \text{ are not in conflict}
\end{cases}
\]

The double integral in the probability of conflict computation can be approximated by a double summation. First a discretized representation of the probability distribution as described in Section IV is sought.

\[
\left[ \left( s_{1}, P_{s_{1}} \right), \left( s_{2}, P_{s_{2}} \right), \ldots, \left( s_{m}, P_{s_{m}} \right) \right], P_{s_{k}} = \text{Prob}(s \leq s_{k})
\]

The mid-points of the discretized path length variable vector are chosen for conflict evaluation.

\[
S_{i} = \left\{ \frac{s_{1} + s_{2}}{2}, \frac{s_{2} + s_{3}}{2}, \ldots, \frac{s_{(m-1)} + s_{m}}{2} \right\}
\]

\[
S_{j} = \left\{ \frac{s_{1} + s_{2}}{2}, \frac{s_{2} + s_{3}}{2}, \ldots, \frac{s_{(m-1)} + s_{m}}{2} \right\}
\]

The function \( P_{conf} \) for computing the probability of a conflict between two takes in three input arguments. The first and second arguments are the cumulative probability distribution functions \( P_{1}(s_{1}) \) and \( P_{2}(s_{2}) \), associated respectively with the path length variables of the first and second aircraft. The third input \( C_{12}(s_{1}, s_{2}) \) is a classification of the conflict for different pairs of the path length variables. The output from the \( P_{conf} \) is the probability of conflict:

\[
P_{conf} : (P_{s_{1}}, P_{s_{2}}, C_{12}) \rightarrow P_{conf}(P_{s_{1}}, P_{s_{2}}, C_{12}) \in [0,1]
\]

Probability of conflict:

\[
P_{conflict} = P_{conf}(P_{s_{1}}, P_{s_{2}}, C_{12}) = \sum_{i=1}^{(m-1)} \sum_{j=1}^{(m-1)} C_{12}(i, j)(P_{s_{1}}(i + 1) - P_{s_{1}}(i))(P_{s_{2}}(j + 1) - P_{s_{2}}(j))
\]
D. Metrics

Trajectory processing results are aggregated into compact and easy-to-use forms suitable for usage by conflict resolution algorithms. The following output parameters are chosen for this purpose:

- **Time of initial conflict**: This is the first instant of time when the probability of conflict crosses a pre-chosen threshold.
- **Duration of conflict**: This is the duration over which the probability of conflict remains above the pre-chosen threshold.
- **Maximum conflict probability**: This is the maximum probability of conflict value assumed by the pair of aircraft.
- **Time to maximum conflict probability**: This is the time at which the aircraft pair assumes the maximum probability of conflict.
- **Time of collision**: This is the first instant of time when the probability of collision crosses a pre-chosen threshold.
- **Maximum collision probability**: This is the maximum probability of collision assumed by the pair of aircraft.
- **Time of maximum collision probability**: This is the time at which the aircraft pair assumes the maximum probability of collision.

VIII. Results

E. Trajectory Prediction

Figure 17 shows the standard deviation of the error predictions as a function of time for different aircraft types using MAESTRO’s Trajectory Prediction module.

![Figure 17. Standard Deviation of Trajectory Prediction Errors](image)

F. Deterministic Conflict Detection Results

MAESTRO has been evaluated using actual DFW surface traffic. Figure 18 shows a block diagram of this evaluation process. The particular data that was used is based on the South Flow configuration. A total of 130 flights...
over a time period of about 1 hr were chosen for the evaluation. The total computational time by MAESTRO is around 40 minutes indicating potential for real-time realization of MAESTRO. The surface traffic data is obtained from real surveillance systems, hence, they capture realistic surveillance errors. In addition to surveillance data MAESTRO requires planner data as well. SARDA planner is currently not implemented at any airports, hence, it is unrealistic to expect actual SARDA operational data. However, for the purpose of current research evaluation a “Surrogate SARDA Planner” is created. The surrogate planner processes the surface traffic data offline and identifies the taxiway routes as well as the time the departure aircraft arrive at the ramp spot. These two pieces of information obtained from offline processing are treated as additional inputs to MAESTRO. The surveillance data is first evaluated using the conflict detection logic to identify “Actual Conflicts” if any that occurred within this data. Since this is real operational data, it is natural to expect zero conflicts. The surveillance data together with the surrogate planner output data is then processed using MAESTRO’s trajectory prediction and conflict detection logic. The output of MAESTRO would be the “Predicted Conflicts.” The difference between the actual and predicted conflicts are characterized as missed alerts and false-alarms. The experiment thus models the following challenges to conflict detection that a computer simulation may not be able to:

- Real Surveillance Errors
- Real Aircraft Operational Uncertainties
- Real Current Day ATC Operational Uncertainties
- Uses No Intent Information Other than Taxiway Route and Ramp Spot Release Time

Table 4 compares the actual and predicted conflicts. It can be seen from this table that there are no actual conflicts and also no missed alerts. Figure 19 shows the evolution of the inter-aircraft separation as a function of time of the taxiway collision predicted by MAESTRO. It can be seen from the figure that the separation reaches a value that is very close to the threshold.

![Figure 18. Conflict Detection Evaluation](image)
Table 4. Actual and Predicted Conflicts

<table>
<thead>
<tr>
<th></th>
<th>Actual # Conflict/Alerts</th>
<th>Predicted # Conflicts/Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taxiway Collisions</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Head-On Collisions</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Runway Incursions</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>RISAs</td>
<td>30</td>
<td>36</td>
</tr>
</tbody>
</table>

Figure 19. Inter Aircraft Separation for Predicted Taxiway Conflict

Figure 20 shows the head-on collision detected by MEASTRO. Head-on collisions occur very rarely. However, it can be seen from Figure 20 that the aircraft were bound to be on the same link with reciprocal headings at the same time. The conflict has not been realized in actual operations because a conflict resolution action was taken by one of the aircraft as seen in Figure 21. Aircraft B stops before the intersection to allow Aircraft A to cross before making the left turn. Thus an imminent head-on collision detected by MAESTRO was avoided.
Figure 20. Predicted Head-On Collision

Figure 21. Head-On Collision Averted

Figure 22 and Figure 23 depict the runway incursion and RISA scenarios detected by MAESTRO. The red and blue dots indicate the positions of the two aircraft at the time of conflict. Again, these scenarios are very close to the definition of the runway incursion and RISA. Both Runway Incursion scenarios (see Figure 22) involve more than one aircraft on the same runway at the same time. Most of the RISAs (see Figure 23) involve departure aircraft (red dots) in the hold pad or on the runway; and crossing aircraft (blue dots) waiting to cross the same runway.
G. Probabilistic Conflict Detection Results

The previous section presented results obtained using actual traffic data and indicated zero missed-alerts and few false-alarms that are actually close conflicts. Deterministic conflict detection is expected to be somewhat robust to trajectory prediction errors when detecting runway incursions and generating RISAs. This is largely due to the conservative assumptions made by the trajectory predictor in Section IV. The trajectory prediction precludes the possibility of missing runway incursions and RISAs because of temporal trajectory prediction errors. However, the same cannot be said about taxiway collisions. Deterministic conflict detection with a fixed separation tolerance...
cannot detect scenarios involving close inter-aircraft separation. Probabilistic conflict detection framework is applied to such scenarios. Figure 24 shows the inter-aircraft separation plot of another pair of aircraft whose minimum separation falls to 302 ft. The threshold for conflict detection is 140 ft. Therefore, deterministic conflict detection could deem this situation as completely conflict free. Probabilistic conflict detection on the other computes the probability of conflict. Figure 25 shows the probability of conflict prediction which indicates as maximum of 35% percent probability of conflict taking into account the trajectory prediction errors. This could be deemed a low-probability conflict which could however be brought to the attention of the flight crew. Figure 26 shows the time-of-conflict prediction based on a 30% probability of conflict threshold.

Figure 24. Inter-Aircraft Separation

Figure 25. Probability of Conflict Prediction

Figure 26. Time of Conflict Prediction

Figure 27 shows the inter-aircraft separation of a pair of aircraft that come very close to each other but not close enough to cross the threshold indicated by the red line. Figure 28 shows the probability of conflict prediction made by MAESTRO. The minimum separation observed in this case is 240 ft and the threshold for conflict detection is 150 ft. Figure 29 shows the time of conflict prediction made by MAESTRO.
IX. Conclusion

The paper develops a comprehensive conflict detection automation system called MEASTRO for NextGen airport surface operations. The approach is shown to take advantage of the intent information resulting from airport operational planners such as SARDA. A new conflict alert called Runway Incursion Situational Alert was formulated to predict Runway Incursion like situations with adequate lead-time. The performance of MAESTRO was evaluated using actual DFW surface traffic and in house closed loop simulations. Preliminary testing indicates zero missed-alerts, few false alarms that are actually close conflict encounters. It was observed that conflicts could be detected with lead-times as early 60 seconds. Future work could involve further evaluation and refinement of the deterministic and probabilistic algorithms.

Acknowledgments

This research has been performed under NASA support through an NRA contract from Ames Research Center. The authors thank Ms. Sandy Lozito, Dr. Yoon-Jung, and other researchers from the Safe and Efficient Surface Operations (SESO) group for their inputs, suggestions, and feedback.


