

**Project Report
ATC-333**

**Surveillance Improvement Algorithms for
Airport Surface Detection Equipment
Model X (ASDE-X) at
Dallas-Fort Worth Airport**

**J. L. Gertz
R. D. Grappel**

20 April 2007

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS



Prepared for the Federal Aviation Administration,
Washington, D.C. 20591

This document is available to the public through
the National Technical Information Service,
Springfield, VA 22161

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

1. Report No. ATC-333		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Surveillance Improvement Algorithms for Airport Surface Detection Equipment Model X (ASDE-X) at Dallas-Fort Worth Airport				5. Report Date 20 April 2007	
				6. Performing Organization Code	
7. Author(s) Jeffrey L. Gertz and Robert D. Grappel				8. Performing Organization Report No. ATC-333	
9. Performing Organization Name and Address MIT Lincoln Laboratory 244 Wood Street Lexington, MA 02420-9108				10. Work Unit No. (TRAIS)	
				11. Contract or Grant No. FA8721-05-C-0002	
12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration 800 Independence Ave., S.W. Washington, DC 20591				13. Type of Report and Period Covered Project Report	
				14. Sponsoring Agency Code	
15. Supplementary Notes This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, under Air Force Contract FA8721-05-C-0002.					
16. Abstract Operational testing of the Runway Status Lights (RWSL) system at the Dallas/Fort Worth (DFW) airport has detected a number of cases where faults in the ASDE-X/DFW surveillance data have led to erroneous operation of the status lights. Among the surveillance problems noted during testing at DFW were: (a) false tracks, (b) track positional jumps to false locations, (c) Mode S track splits, (d) ATRBS track splits, (e) invalid Mode C altitudes, (f) invalid track velocities, and (g) spurious Mode 3/A 0607 _g code tracks. The RWSL surveillance improvement algorithms package described in this document is placed between the ASDE-X/DFW surveillance data source and the RWSL safety logic. The surveillance improvement algorithms perform a variety of reasonableness and consistency checks on the input data and set validity flags and report status values for each input report which are then passed on to the RWSL safety logic. These flags and status values allow the RWSL to ignore erroneous reports and to avoid using questionable report components in the subsequent RWSL logic. This document illustrates the performance of the RWSL surveillance improvement algorithms package with examples from DFW analysis. It is shown that the RWSL surveillance improvement algorithms package substantially reduces the impact of the known ASDE-X/DFW surveillance anomalies on the performance of the RWSL safety logic. The RWSL surveillance improvement algorithms package may also host future algorithms necessary to mitigate further problems that might be detected in the surveillance data.					
17. Key Words			18. Distribution Statement This document is available to the public through the National Technical Information Service, Springfield, VA 22161.		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 40	22. Price

EXECUTIVE SUMMARY

The Runway Status Lights (RWSL) system accepts, in addition to ASDE-3 surface primary radar data, a stream of surveillance inputs from the prototype ASDE-X combination of sensors (including Mode S multilateration, ATCRBS multilateration, ASR-9 primary/secondary radar, and ADS-B data sources) under evaluation at the Dallas/Fort Worth (DFW) airport. Fused surveillance data from these sensors is used to detect potentially hazardous conditions and to operate warning lights—Takeoff Hold Lights (THLs) and Runway Entrance Lights (RELs)—placed at strategic locations on the airport surface. However, if the input ASDE-X/DFW surveillance data contains anomalies and inaccuracies, the RWSL safety logic outputs may be incorrect.

Operational testing of the RWSL system at the Dallas/Fort Worth (DFW) airport has detected a number of cases where faults in the ASDE-X/DFW surveillance data have led to erroneous operation of the status lights. Among the surveillance problems noted during testing at DFW were:

- a. False tracks
- b. Track positional jumps to false locations
- c. Mode S track splits
- d. ATCRBS track splits
- e. Invalid Mode C altitudes
- f. Invalid track velocities
- g. Spurious Mode 3/A 0607₈ code tracks

Algorithms that address each of these problem areas are described in detail in this document.

The RWSL surveillance improvement algorithms package described in this document is placed between the ASDE-X/DFW surveillance data source and the RWSL safety logic. The surveillance improvement algorithms perform a variety of reasonableness and consistency checks on the input data and set validity flags and report status values for each input report which are then passed on to the RWSL safety logic. These flags and status values allow the RWSL to ignore erroneous reports and to avoid using questionable report components in the subsequent RWSL logic.

This document illustrates the performance of the RWSL surveillance improvement algorithms package with examples from DFW analysis. It is shown that the RWSL surveillance improvement algorithms package substantially reduces the impact of the known ASDE-X/DFW surveillance anomalies on the performance of the RWSL safety logic. The RWSL surveillance improvement algorithms package may also host future algorithms necessary to mitigate further problems that might be detected in the surveillance data.

Since these algorithms have been in evaluation mode, no light activations caused by false tracks have been observed, and prior known false activations have been eliminated when subjected to the new software. Although negative repercussions are always possible when tracks are eliminated, a number of checks applied before a track is called false—speed, altitude, ASDE-3 concurrence, ASR-9 and ADS-B confirmation—have reduced the probability so low that no loss of real tracks in the critical safety logic regions have yet been noted in testing.

TABLE OF CONTENTS

Executive Summary	iii
List of Illustrations	vii
1. INTRODUCTION	1
2. ALGORITHMS	3
2.1 False Track Rejection	4
2.2 Multiple Track Resolution	16
2.3 Identification of Invalid Mode C Altitudes	19
2.4 Rejection of Invalid Track Velocities	20
2.5 Rejection of '0607' Spurious Tracks	20
3. SAMPLE RESULTS	23
4. SUMMARY	31
ACRONYMS	33

LIST OF ILLUSTRATIONS

Figure No.		Page
1	RU Mode S grid cell coverage for DFW.	23
2	Targets declared false (24 hours) at DFW.	24
3a	Multilateration reports received for Mode S address A64C88.	25
3b	Tracks (real and false) declared by ASDE-X/DFW from above reports.	25
4a	Example of ASDE-X/DFW track jump, reports input to tracker.	26
4b	Example of ASDE-X/DFW track jump, positions output by tracker.	27
5a	Example of ASDE-X/DFW track split due to Mode S Transponder failure.	28
5b	Example of ASDE-X/DFW track split due to ASDE-X/DFW Fusion failure.	28
6	Invalid velocity estimates for slow moving aircraft.	29

1. INTRODUCTION

This report describes the set of ASDE-X surveillance improvement algorithms developed as part of the Runway Status Lights (RWSL) system. This system attempts to prevent aircraft collisions on the aircraft surface, particularly on runways where aircraft speed would make such accidents catastrophic, by illuminating Takeoff Hold Lights (THLs) and Runway Entrance Lights (RELs) to warn aircraft of impending dangers should they proceed. The inputs to the system are surveillance reports from the surface primary radar ASDE-3 sensor and the beacon ASDE-X sensors (multilateration, ASR-9, and ADS-B). These surveillance reports are fused by RWSL into system tracks, which are then used to drive the safety logic.

The ASDE-X system configuration assumed in this report is the specific one currently undergoing evaluation testing at Dallas/Fort Worth International Airport (DFW) (hereafter called the ASDE-X/DFW). The ASDE-X/DFW includes 13 multilateration receiver units (RUs), has connections to two ASR-9 terminal radars, but does not have a surface movement radar component. Other ASDE-X systems may have different sets of input sensors, and thus different fusion and tracking algorithms. In that case, the surveillance improvement algorithms presented herein may require modifications. However, the multilateration system whose reports are the key element driving the surveillance improvement package is typical of the expected production systems. The actual ASDE-X/DFW system is ASDE-X build ASDEX 5.0.7.1.1, which includes Build MDT_TP_MS DP 5.1.6, released 22 November 2005, and its interfaces are defined in the ASDE-X IDD version 8.0, dated 7 July 2005.

Operational testing of the RWSL system at the DFW airport has detected a number of cases where faults in the ASDE-X/DFW surveillance data have led to erroneous operation of the status lights. Among the surveillance problems noted during testing at DFW were:

- a. False tracks
- b. Track positional jumps to false locations
- c. Mode S track splits
- d. ATRBS track splits
- e. Invalid Mode C altitudes
- f. Invalid track velocities
- g. Spurious Mode 3/A 0607₈ code tracks

These anomalies and transients are caused by:

- a. Problems in the particular avionics of a given aircraft,
- b. Problems in the surveillance functions of a particular sensor, or
- c. Problems in the ASDE-X/DFW data merging and tracking functions.

Algorithms that address each of these problem areas are described in detail in this document.

The surveillance improvement algorithms package acts as a filter between the input ASDE-X/DFW surveillance data input stream and the RWSL tracker and safety logic, and is intended to correct surveillance errors existing in the ASDE-X/DFW system that have created invalid reports. Various types of surveillance anomalies and transient events from the input surveillance are detected and flagged, so that they do not impact subsequent RWSL processing. This report describes the particular problem areas dealt with by the surveillance improvement algorithms package, and the logic and implementation used to detect and correct each specific problem. All algorithm parameters presented in this report have their nominal values specified; these values were derived from analysis of anomalous cases detected in recordings of RWSL surveillance input data from DFW. Parameter values for other airports may differ due to variations in airport geometry and RU coverage.

Since these algorithms have been in evaluation mode, no light activations caused by false tracks have been observed, and prior known false activations have been eliminated when subjected to the new software. Although negative repercussions are always possible when tracks are eliminated, a number of checks applied before a track is called false—speed, altitude, ASDE-3 concurrence, ASR-9 and ADS-B confirmation—have reduced the probability so low that no loss of real tracks in the critical safety logic regions have yet been noted in testing.

2. ALGORITHMS

The main input to the surveillance improvement algorithms package is a stream of ASDE-X/DFW reports from System Track Messages 0x25, transmitted from the MSDP to the MDT on logical port MF, using IP multicast on address 224.50.100.2 port 3920. Each input report contains a track identifier number (will be 0 if report is uncorrelated), a reference ID, smoothed position and velocity vectors if the report is tracked, time of observation, and the raw surveillance data from the source that generated this report (Mode S multilateration, ATRBS multilateration, ASR-9, ADS-B). For format and field definitions, refer to Section 3.9.3.30 in the ASDE-X IDD version 8.0, dated 7 July 2005.

Additional inputs to the surveillance improvement algorithms package include a database for all the multilateration “Receiver Units” (RUs) containing their geographic locations and whether they are configured to be active. A second RU database contains a coverage map dividing the airport surface into a set of Cartesian bins. Each bin entry contains for each RU the probability that this particular RU provides coverage for this geographic region (i.e., how likely is it that an aircraft in this region would be seen by the RU). The RU data is used by several of the range-confirmation and scoring functions described below.

The main output of the surveillance improvement algorithms package is a report “validity” value for each tracked input report. This validity value indicates whether the report’s position is considered to correctly represent the aircraft location (as opposed to a multipath-induced false position), and hence acceptable for processing by the RWSL tracker and safety logic. There are four defined report validity values in the surveillance improvement algorithms package:

- 0: False report, to be immediately rejected
- 1: Probably false report, to be rejected unless supported by fused ASDE-3 report
- 2: Probably real report, to be accepted unless ASDE-3 indicates no target exists at the ASDE-X/DFW report position
- 3: Real report, to be accepted

Track drop messages are also processed to determine whether they should be considered valid such that the RWSL system track can be deleted, or invalid and the RWSL system track must be maintained as more reports are expected under a different track number (see Section 2.2.4 below).

The surveillance improvement algorithms package also generates validity flags for the velocity and altitude fields found in ASDE-X/DFW reports. The logic used to compute the flag for track altitude is described in Section 2.3 below, while the logic used to compute the flag for track velocity is described in Section 2.4 below. Finally, the surveillance improvement algorithms package may generate a “cross-reference” track number for reports involved in the multiple track resolution algorithms described in Section 2.2 below.

The primary local data structure in the surveillance improvement algorithms package is a track file with an entry for each active track in the system. The track file is maintained using the C++ “map” package. All the state information computed for each system track in the surveillance improvement algorithms package is maintained in its track file entry.

An additional local data structure in the surveillance improvement algorithms package is the “code map.” This set of data structures is accessed by a report via its Mode S code (if it has one), Otherwise via its discrete Mode 3/A ATRBS code (a Mode 3/A code of 1200₈ is not unique to a given aircraft, so to avoid confusion, such reports are not mapped to a code map structure entry). The code map is maintained using the C++ “map” package. State information about an aircraft code is maintained in the code map so that information from tracked and untracked reports from the same aircraft can be connected.

The surveillance improvements package expresses horizontal distances in units of meters and vertical distances (altitude) in units of feet. Horizontal speeds are expressed in units of knots and vertical rates are expressed in units of feet-per-second. Times are expressed in units of seconds. For clarity, thresholds used in the various algorithms are given here using their empirically derived default value.

2.1 FALSE TRACK REJECTION

2.1.1 Algorithm Overview

The first step of the surveillance improvements function is to obtain the track file entry for this input target report’s track number (track identifier number 65535 is used to signal an uncorrelated report). If the track entry did not already exist, it is created now and initialized. If the time since the track file entry was last updated exceeds 1000 seconds (indicating that the track drop was somehow missed), then the track file entry is re-initialized.

The time value in certain ASDE-X/DFW surveillance records (e.g., track drop reports) may be out of sequence with respect to the normal report updates in the input stream. The time value used for these reports is reset internally to the time of the most recent normal ASDE-X/DFW report. Input reports for system test are immediately declared real and receive no further processing.

Once each day (at midnight), a garbage collection function is run to remove code map entries that are too old and should be deleted. If any particular code map entry in the database has not been updated in at least 100 seconds, it is deleted here.

Track drop reports are flagged real or false when received. False drop cases correspond to certain split track situations (see Section 2.2) and are used as a signal to indicate to RWSL that the corresponding system track should be maintained. Tracks dropping prematurely and tracks supporting other tracks (i.e., those with inverse cross-references) are the tracks labeled as false drops.

Next, a position reasonableness test is performed comparing the report's raw position with its smoothed track position. Reports that have been smoothed too far by the ASDE-X/DFW tracker are highly suspect, and hence are not used in the multilateration processing algorithms described below. The reports that are declared false (validity = 0) and not processed further are those satisfying all of the following:

- a. Smoothed distance difference greater than 200 meters
- b. Time since last track update less than 5 seconds
- c. Report is a multilateration report, or the time since the last multilateration update is less than 5 seconds

The next processing step is to perform the altitude and velocity validity checking described in Sections 2.3 and 2.4 below. Flags and validity times for velocity, altitude, and Mode S and ATRBS codes are stored in the track file for later use.

If the input report has not been declared false as yet, processing continues with the functions described in Sections 2.1.3 through 2.1.8 below. The ultimate result is a validity value for this input report. Finally, the track split merging algorithms described in Section 2.2 are performed. The final outputs of the surveillance improvement algorithms, returned to the caller, are:

1. report's validity value
2. velocity validity flag
3. altitude validity flag
4. track number to use for the report (will differ from the report's track field if track merging has occurred)

2.1.2 RU Coverage Statistics

The multilateration system uses a constellation of receiving units (RUs) on the airport surface to provide the multiple time-of-arrival information required to generate a position report. In order to provide coverage of all locations on the airport, a large number of such RUs are sited, each of which sees some, but not all, of the airport. Thus the set of RUs creating a particular report is a function of aircraft position, and reports at any given position will in general indicate the same or similar set of RUs. On the other hand, if an RU has multipath in the receiving path, causing a false multilateration position to be computed, the set of RUs in the report may not match the set known to apply to that position. The false report detection algorithm employs a set of RU coverage statistics to be able to detect such events. RUs not expected for a location, and RUs missing for a report at that location, are used to create a "falseness" score as described in Section 2.1.5 below.

The coverage statistics database for the RWSL multilateration RUs is currently gathered by a stand-alone program, although a real-time version is under development. This program reads and processes ASDE-X/DFW surveillance files covering a significant period of time (several weeks at least, including periods of time in all runway configurations). The RU coverage statistics are generated for a Cartesian grid extending plus or minus 7500 meters from the airport center (the area over which multilateration input data is expected to occur). Currently, the program generates RU statistics for two grid sizes: 30 meters square and 100 meters square (although only the 30-meter grid data is presently used in the surveillance improvement algorithms). The following statistical counts are computed for each grid cell:

- a. Number of times that an input multilateration report was located in this grid cell (“hits”),
- b. Number of times that an input report with range confirmation data (see Section 2.1.3) was located in this grid cell (“confirms”), and
- c. Number of times that each particular active RU in the system contributed to the multilateration solution of an input report in this grid cell (“ruhits”).

Uncorrelated input reports are not counted in the statistics, as they are suspect, nor are reports from system test tracks or CPMEs. Reports labeled as false are counted, however, as deleting data from reports incorrectly called false could lead to an increase in such false declarations, possibly causing a light activation to be missed. Fortunately, as shown in Section 3, false reports are rare and will not skew the statistics significantly.

A special text line is written to the RU coverage statistics output file as a header to indicate the extent of the grid cell array (7500 meters) and the size of grid cell in use (30 or 100 meters) in this particular data file. This special text line is indicated by a period (“.”) character at the start of the line. The output format for the RU coverage statistics files next includes a text line for each grid cell. The line begins with six values as follows: row index of cell, column index of cell, x-location of cell in meters, y-location of cell in meters, “hits” count for cell, and “confirms” count for cell. The remainder of each text line includes the “ruhits” counts for this cell for each of the adapted RUs in the RWSL system (up to a maximum of 32).

It is useful to know how well the RU coverage statistics are converging as more and more target reports are processed. The test currently used is to compute the total number of grid cells with sufficient “hits” to allow for RU scoring (see Section 2.1.5 below). The number of “hits” for each grid cell and its 8 nearest neighbor cells is computed. If the number of such “hits” exceeds 500, then this grid cell is counted as score-able. The total count of such score-able grid cells is a measure of the completeness of the RU coverage statistics database, and when the count no longer changes significantly the statistics have reached a stable state. In addition, a map of the score-able grid cells can be generated to verify that coverage includes all runways and important taxiways. At DFW, 44 days of data was required to reach a stable state.

In general, due to the difference in Mode S and ATCRBS RU coverage and accuracy statistics, both a Mode S and ATCRBS RU database would be generated. At DFW the large majority of airport operations are due to commercial aircraft that are equipped with Mode S transponders, so insufficient ATCRBS data was available to generate its database during the development effort (more than a year would have been required). Thus the RU statistics database described in this report used only Mode S surveillance data, and was then applied to both Mode S and ATCRBS multilateration reports. In an environment where significant ATCRBS traffic was expected (sufficient to develop a valid RU coverage set in a time comparable to that for Mode S), a distinct ATCRBS-only RU database would be computed and used for ATCRBS tracks. In other environments, such as DFW, the Mode S database would be applied until sufficient ATCRBS data was processed.

2.1.3 Range Confirmation Tests

Most multilateration reports are derived from reception of Mode S squitter replies by the RU array. However, three other types of replies also generate multilateration reports: (1) Mode S DF-4 altitude replies, (2) Mode S DF-5 identity replies, and (3) ATCRBS replies. These replies are all received in response to interrogations transmitted by one of the RUs. If the interrogating RU receives the reply, that RU knows both the time of transmission and the time of reception. Hence the range to the aircraft can be computed directly with an error determined by the errors in time measurement and the transponder turnaround delay error.

The range confirmation test algorithms are employed to validate multilateration position reports (which may be spurious) generated by any of these replies by comparing the multilateration report's range from the transmitting RU to the range derived from reception time. Agreement indicates a likely valid report, while disagreement is a symptom of a false report. Since only two dimensional multilateration ranges are reliable, the test is only performed for targets on the airport surface (Mode C altitude less than 1000 feet for DFW).

The first step in range confirmation test processing is to extract the multilateration information from the input Mode S and ATCRBS target reports. Each such target report's data includes:

- a. Flag indicating whether range confirmation data is present in this report,
- b. RU that transmitted the aircraft's transponder interrogation (if flag set),
- c. That RU's measured range to the aircraft (if flag set), and
- d. Multilateration position for the aircraft (if flag set).

The range confirmation processing computes the distance from the multilateration report's position (if present) to the indicated RU. This distance is then compared to the RU's measured range to the aircraft. The difference in position (which should be very small) is used in the subsequent steps of range confirmation processing. A difference threshold value of 100 meters is used to pass or fail the report's

multilateration position. This value is dependent upon multilateration positional accuracy and transponder turnaround delay errors; the 100-meter value was found empirically to apply at DFW.

The range confirmation test algorithms maintain a history of successful and failed confirmations for each track. The relevant fields are:

1. Number of successful confirmations
2. Time of most recent successful confirmation
3. Position of most recent successful confirmation
4. Number of failed confirmations
5. Time of most recent failed confirmation
6. Position of most recent failed confirmation
7. If the last confirmation was a success, number of successive successful range confirmations
8. If the last confirmation was a failure, number of successive failed range confirmations

Whenever either successive field (7 or 8) reaches 5, indicating a trend, the number of the other type of confirmations (4 or 1, respectively) is reset to 0.

One complication to the algorithms is that this history cannot be maintained completely by track. If the report range confirmation data does not match the multilateration position, the ASDE-X/DFW tracking algorithm may leave the report as uncorrelated. Thus, if only reports for a given track were examined, many bad range confirmation events would be lost. The range confirmation algorithms thus make use of the “code map” database as well as the track file. Each Mode S and discrete ATRBS code has an entry, the entry containing the positions and times of the most recent uncorrelated reports with successful and failed report range confirmation tests.

Every input report with range confirmation information is processed by this algorithm. There are four cases to consider: the given report may or may not have a correlating track number, and the given report will have either passed or failed the range confirmation test.

If the input report does not have a correlating track number, but does have a code map entry, then the range confirmation data for this report must be saved in its code map entry instead of a track file entry. The data is saved in the section of the code map entry corresponding to the test result, either successful or failed. Then, when an input report with a track identifier number is received for this code map entry, checks are made to determine whether to copy the code map entry data into the track file (tests performed separately for successful and failed sub-entries). This transfer is undertaken if the code map entry is less than 5 seconds old (indicating the information is current), and the distance between the current report’s position and the entry position is less than 100 meters (indicating that the information applies to the current track location). In either case, the code map entry is then cleared. On the other hand,

if the input range confirmation report has a track identifier number, the data is simply stored directly into the track file entry, with each applicable field being updated.

The main reason for collecting range confirmation data is to determine what bias if any to apply to the threshold “mnscore” used in the RU report scoring process described in Section 2.1.5 below. This threshold is the minimum average penalty score required to call a report false. If successful range confirmations exist for the track, the track is more likely to be real, and the threshold is raised; failed confirmations conversely indicate a more likely false track, and the threshold is reduced. The default value for “mnscore” is 2.

If the track file entry has only failed range confirmations, or if it has at least 3 failed range confirmations and only a single successful confirmation, and the distance between the input report position and the track’s most recent failed range confirmation position is less than 300 meters (so that the confirmations are still relevant), then “mnscore” is reduced to 1.5. Conversely, if the track has only successful range confirmations, or there have been at least 3 successful range confirmations and only a single failed confirmation, and the distance test is passed, “mnscore” is raised to 2.5.

Range confirmation data is also used in the scoring adjustment Section 2.1.6 below to give a second chance to false reports that have a favorable successful confirmation history. This aspect of the algorithm is discussed in that section.

2.1.4 Track Positional Jumps

As an aircraft moves on the airport surface, the aircraft/RU geometry changes. If the change results in multipath in the signal path to an RU, the resulting multilateration solution position can undergo a significant jump to a false location. The track jump algorithm in the surveillance improvement algorithms package is used to detect cases where this jump has occurred, but the new false position is close enough to the true one that the ASDE-X/DFW tracker maintains the same track number. When such a surveillance positional anomaly is detected, the track jump algorithm forces a re-initialization of the track file entry for this track so that the prior history of real reports does not prevent new reports from being declared false.

The threshold for excessive track movement is a function of the track’s most recent velocity. If the track’s velocity is greater than 30 knots then the movement threshold is set to 300 meters – it is set to 100 meters for slower tracks. (Note: the track jump algorithm is intended for aircraft on the surface. The most recent non-zero altitude seen for the track is maintained in its track file entry in order to determine if the aircraft is airborne or not.)

The track jump algorithm stores three previous aircraft x-y positions in the track file entry: (1) the most recent position declared “reliable,” (2) the previous position declared “reliable,” and (3) the most recent position declared “unreliable” (too far from the track – a potential “jump”). The track entry also maintains a count of the number of “reliable” position updates prior to the jump (“jhits”) and a count of the number of position updates that are consistent with the jump position (“failcount”).

The jump logic is applied to tracks that are on the ground (i.e., its most recent Mode C altitude is no greater than 1000 feet at DFW), are slow moving (i.e., its most recent velocity is no greater than 60 knots), and have at least 2 current reliable position updates (“jhits”). The value of “jhits” is incremented when the distance between the current report position and the previous one is less than half the distance threshold defined above. In this case, these 2 reports are saved as the “reliable” positions and the “failcount” value is set to 0.

The following jump-seeking logic is performed on each altitude and speed eligible new track report, and whose track “jhits” value is 2 or greater. Three distances are computed:

1. Between the report and the most recent reliable position,
2. Between the report and the previous reliable position, and
3. Between the report and the most recent “jump” position (if one exists).

If the “failcount” is zero (indicating no “jump” position exists) and both distances (1) and (2) exceed the movement threshold, the start of a potential track jump condition is declared. The “failcount” value is initialized to 1 and the report position is stored as the track’s “jump” position.

If the “failcount” is greater than zero, distances (1) and (2) exceed the movement threshold, and distance (3) is less than half the movement threshold (indicating report data consistent with the “jump”), then the track’s “failcount” is incremented. Should the value of “failcount” reach 3, the track jump is considered confirmed and the track file entry is re-initialized. Otherwise, if the distance (1) and (2) tests are satisfied but the distance (3) is too large for a consistent jump, the track jump logic restarts for a different “jump” using the new report.

If during a “jump” situation the distance (1) is found to be less than half the movement threshold, the track is considered to have returned to the “reliable” position. In that case, the “jhits” value is incremented and the current report position is copied to the most recent reliable position in the track file entry. Jump logic then will restart should a new inconsistent measurement be recorded.

Finally, to prevent false reports that have initiated a “jump” event from being called real because the “jump” has not yet been confirmed, a later component of the track jump algorithm is performed after the report scoring function described below in Section 2.1.5 is completed. If the track’s “failcount” is greater than zero, indicating a “jump” event is in progress, or its “jhits” value is zero, indicating lack of “reliable” data, the report is declared to be “false” if its RU score satisfies the relevant criterion:

- a. “scoresign” = +1 and “score” greater than or equal to 2
- b. “scoresign” = -1 and “score” greater than or equal to 3
- c. “scoresign” = 0 and “newscore” greater than or equal to 3

2.1.5 Report Scoring

The report scoring algorithm in the surveillance improvement algorithms package computes a penalty score for a multilateration report based on how much the RUs used to compute the report's position differ from the expected RU set for that location. Both missing expected RUs and existence of unexpected RUs contribute to the score. The precise details of the scoring algorithm, and the parameter settings employed, are a function of the airport RU constellation. This report presents the default values for DFW. Values for other airports would require analysis of recorded data from that airport.

To determine this ideal set of RUs, the airport surface is subdivided into a Cartesian grid, and each grid cell location maintains statistics built up over time of which RUs were found in the report every time an aircraft is detected in that grid cell, as discussed in Section 2.1.2 above. The report scoring function returns three values denoted "scoresign," "score," and "newscore." The "scoresign" variable indicates which scoring case is used, and has three possible values:

- +1: Sufficient trials exist in the report's grid cell, and the largest possible penalty score is sufficiently large
- 1: Sufficient trials exist in the report's grid cell, but the largest possible penalty score is not sufficiently large
- 0: Insufficient trials exist in the report's grid cell

The "score" variable is an integer value giving the penalty score computed from the number of missing expected RUs and existing unexpected RUs. (Note: lower scores are considered better than higher scores in this algorithm). The "newscore" variable is a floating-point value that computes an alternative penalty scoring value that is used if there is insufficient RU data to compute "score" reliably.

The report-scoring algorithm performs a search through the RU grid cells plus or minus a 30-meter measurement-error tolerance in each axis around the report's position. The weighted fraction of hits for each RU is computed (this fraction is the number of trials for which that RU was found, divided by the total number of trials). The "highest possible" score variable ("posscore") is then computed. The "ascore" variable counts cases where the RU shouldn't have detected the aircraft (based on history) but did, while the "mscore" variable counts cases where the RU should have detected the aircraft but did not.

Each RU declared operational in the current system adaptation is checked in turn. If the weighted fraction of hits for this RU is less than 0.15, the "posscore" is incremented by 2 as an available penalty for an RU that shouldn't detect this report. If the weighted fraction of hits for this RU is greater than or equal to 0.8, the "posscore" is incremented by 1 as an available penalty for an RU that should detect this report. The penalty is smaller for a missing RU because missed detections often exist even when the RU receives a signal due to decoding failures.

If a report indicates that a given RU was used in its multilateration solution, and the weighted fraction of hits for that RU is too small (less than 0.15), then “ascore” is incremented. If the weighted fraction of hits is less than 0.5, then the alternative “newscore” is penalized by a value between 0 and 2 given by:

$$\text{Penalty}_{\text{false detection}} = 4 * (0.5 - \text{hit fraction})$$

Similarly, if the report indicates that a given RU was not used in the multilateration solution, and the weighted fraction of hits is too large (greater than or equal to 0.8), then “mscore” is incremented. If the weighted fraction of hits is greater than 0.5, then the alternative “newscore” is penalized by a value between 0 and 1 given by:

$$\text{Penalty}_{\text{missed detection}} = 2 * (\text{hit fraction} - 0.5)$$

Finally, the value of “score” is given by (2*ascore + mscore), while the value of “newscore” is rounded to the nearest integer. As explained above, the scoring penalty for missing a detection is one half that for a likely false detection.

There are three scoring cases to consider, based on the total number of statistical samples for this grid cell and the “possscore.” If there were at least 500 samples and the “possscore” is at least 7, then “scoresign” is set to +1 and “score” is used for the penalty. On the other hand, if there were enough samples but the value of “possscore” was too small, then “scoresign” is set to -1 and the penalty is set to the alternative “newscore” value. Finally, if there were insufficient samples, then “scoresign” is set to 0 and “newscore” again used for the penalty. For the case that applies, the number of reports for that case is incremented in the track file entry and the track’s total penalty score for the case is incremented by the report’s penalty.

If the “scoresign” value is nonzero (indicating that there were sufficient statistics in the current report’s RU grid cell), a check is made to see how many sequential scoring cases exist that were nonzero. If the count reaches 5, the scoresign 0 cases are no longer required for the track (they have low validity), and hence the count of “scoresign” 0 and its corresponding score sum are reset to zero in the track file entry.

The total number of scoring updates on this track, summed over each of the three cases, is computed. Each of the scoring cases is then checked in priority order. If when a scoring case is checked it has at least 25 percent of the total number of track updates, it is used to set the scoring type variable “ruf.” The possible values of “ruf” are:

- 1: cases ₊₁ > 25% of total and score ₊₁ < (mnscore * cases ₊₁)
- 2: cases ₋₁ > 25% of total and score ₋₁ < ((mnscore+1) * cases ₋₁)
- 3: cases ₀ > 25% of total and score ₀ < ((mnscore+1) * cases ₀)
- 0: first case to satisfy 25% rule failed its scoring test

If “ruf” is zero, indicating that the report is about to be labeled “false,” then a scoring check is made to determine if the report itself scored as “real.” The test applied corresponds to the value of “scoresign” for the report:

$$\text{“scoresign”} = +1: \text{“score”} < \text{“mnscore”}$$

$$\text{“scoresign”} = -1: \text{“newscore”} < (\text{“mnscore”}+1)$$

$$\text{“scoresign”} = 0: \text{“newscore”} < (\text{“mnscore”}+1)$$

If the selected test passes, then it appears that the scoring might be labeling this report “false” when it should be “real.” The count of consecutive successful RU scoring tests in the track file entry is incremented and the count of consecutive failed RU scoring tests is reset to zero. If the count of consecutive successful RU scoring tests reaches 5, then the track file entry scoring values are set to make the track “just real”:

$$\text{score}_{+1} = (\text{mnscore} * \text{cases}_{+1}) - 1,$$

$$\text{score}_{-1} = ((\text{mnscore}+1) * \text{cases}_{-1}) - 1,$$

$$\text{score}_0 = ((\text{mnscore}+1) * \text{cases}_0) - 1.$$

If “ruf” is nonzero, indicating that the report is about to be labeled “real,” then a scoring check is made to determine if the report itself scored as “false.” The test applied corresponds to the value of “scoresign” for the report:

$$\text{“scoresign”} = +1: \text{“score”} \geq \text{“mnscore”}$$

$$\text{“scoresign”} = -1: \text{“newscore”} \geq (\text{“mnscore”}+1)$$

$$\text{“scoresign”} = 0: \text{“newscore”} \geq (\text{“mnscore”}+1)$$

If the selected test passes, then it appears that the scoring might be labeling this report “real” when it should be “false.” The count of consecutive failed RU scoring tests in the track file entry is incremented and the count of consecutive successful RU scoring tests is reset to zero. If the count of consecutive failed RU scoring tests reaches 5, then the track file entry scoring values are set to make the track “just false”:

$$\text{score}_{+1} = (\text{mnscore} * \text{cases}_{+1}) + 1,$$

$$\text{score}_{-1} = ((\text{mnscore}+1) * \text{cases}_{-1}) + 1,$$

$$\text{score}_0 = ((\text{mnscore}+1) * \text{cases}_0) + 1.$$

If none of the scoring checks is passed, the report is being properly labeled, and the counts of consecutive RU scoring (both successful and failed) in the track file entry are reset to zero.

The last step of the scoring algorithm is to map the “ruf” value into a preliminary validity decision. If the track is immature (5 or fewer updates), then the mapping is done using the following table:

“ruf”	Validity State
0	False (0)
1	Real (3)
2	Probably Real (2)
3	Probably False (1)

Mature tracks (6 or more updates) use the following table instead:

“ruf”	Validity State
0	False (0)
1	Real (3)
2	Real (3)
3	Probably Real (2)

2.1.6 Score Adjustment Cases

If the input report has not been declared “real” (validity value of 3) by report scoring (Section 2.1.5 above), some special-case tests are performed that may promote the report to a higher validity. These tests consider track speed, altitude history, number of successful range confirmations, and correlations to ASR-9 or ADS-B reports.

False reports are caused by multipath in the aircraft-to-RU signal path caused by intervening structures. Not surprisingly, given airport geometries, the aircraft producing false reports are usually in the gate areas. Thus, fast moving tracks are rarely false. The adjustment rules using this fact are as follows:

1. If the track speed exceeds 60 knots, its non-real reports (0,1, or 2) are converted to “real” (3)
2. If the track is mature, and its speed exceeds 30 knots, its non-real reports (0,1, or 2) are converted to “probably real” (2)
3. If in addition for case (2), the track has successful range confirmation and no failed range confirmations, its reports are further promoted to “real” (3)

Aircraft not on the surface rarely produce reflections (they are above the reflecting structures), so if the track history includes reports with altitude above 1000 feet at DFW, the track's non-real reports (0, 1, or 2) are converted to "real" (3). Of course, the track may have jumped to a false position once it is on the surface. That is why the jump logic of Section 2.1.4 can override this decision.

Finally, the mechanisms that produce false ASR-9 or ADS-B reports are totally different from the mechanism for multilateration reports. Thus, if a track has current ASR-9 or ADS-B updates that correlate in position with the multilateration reports, any non-real tracked reports (0,1, or 2) are changed to "real" (3). This information is contained in the "hasASR9" and "hasADSB" fields of the track file (see Section 2.1.8 below for their calculation algorithm).

2.1.7 Post Fusion State Resolution

A final component of the surveillance improvement algorithms package is performed after the ASDE-X/DFW report has had an opportunity to fuse with an ASDE-3 report. This component of the process converts the intermediate "probably false" and "probably real" report status values from the earlier stages of the algorithms to their final "real" or "false" values. This decision employs two tests: (1) whether there is ASDE-3 coverage at the report's location, and (2) if so, if an ASDE-3 report was fused to the ASDE-X/DFW report. A "probably false" report is promoted to "real" status only if it fuses with an ASDE-3 report; it is demoted to "false" status if no ASDE-3 coverage exists or if coverage exists but no fusion occurs. A "probably real" report is demoted to "false" status if it is in a region where ASDE-3 coverage exists but no ASDE report fused with it. The "probably real" report is promoted to "real" status if either (1) it was fused to an ASDE-3 report, or (2) it is not in a region with ASDE-3 coverage. Note: the regions of ASDE-3 coverage are specified in the system configuration file as a set of overlapping polygons on the airport surface.

The final step in the real/false report declaration process is the addition of hysteresis to keep a suspected ASDE-X/DFW false track from intermittently having reports output as real. Each such switch to the "real" state would cause the RWSL system display of this track to "flicker on," and cause the subsequent RWSL safety logic to process the track. To implement this process, every track maintains counts of the number of successive reports declared "real" and successive reports declared "false." If the "false" count reaches the threshold value 3, all subsequent "real" reports are converted to "false" until there have been at least 3 successive "real" reports for this track. At that point, the "false" count is reset to 0, and the process begins again.

2.1.8 Non-Multilateration Reports

In addition to multilateration reports, ASDE-X/DFW processes and tracks ASR-9 and ADS-B inputs. Since RU statistics do not apply to these reports, a different method of detecting false reports is employed. In addition, these reports must be processed to determine whether they validate multilateration

positions for tracks containing both multilateration and non-multilateration data. This processing sets the “hasASR9” and “hasADSB” flags referenced in Section 2.1.7 above.

This processing only applies to non-multilateration reports correlated to tracks having prior multilateration reports; reports in non-multilateration-only tracks are declared “real.” Non-multilateration reports in mixed tracks are declared “false” if:

1. The time since the last multilateration report is less than 5 seconds, and
2. The x-y distance from the last non-jump multilateration report is at least 300 meters.

All other ASR-9 and ADS-B reports are declared real.

For an ASR-9 report to confirm a mixed track (setting the “hasASR9” flag “true”), the report must satisfy time, distance, and code conditions relative to the multilateration data:

1. Time since the last multilateration report is less than 5 seconds, and
2. x-y distance from the last non-jump multilateration report is no more than 100 meters, and
3. Last confirmed ASR-9 Mode 3/A code is the same as the last confirmed multilateration Mode 3/A code.

For an ADS-B report to set the “hasADSB” flag, conditions 1 and 2 are required; Mode S codes will always agree for any given track supported by ADS-B data.

2.2 MULTIPLE TRACK RESOLUTION

2.2.1 Algorithm Overview

The surveillance input tracker has a penchant for forming two or more tracks on the same aircraft. The three most common occurrences are:

1. Multiple tracks caused by a defective Mode S transponder emitting two different Mode S addresses alternately
2. Multiple tracks caused by the failure to fuse tracks from different surveillance sources, typically ASR-9 and multilateration
3. Multiple sequential tracks when the surveillance source tracker prematurely terminates one track and initiates another one with a different number

When multiple tracks exist at the same time, as in cases 1 and 2, RWSL will form two system tracks on the aircraft. This will obviously cause confusion for the RWSL safety logic, including the prospect of incorrect light operation. In case 3, if there is no ASDE-3 track for the aircraft, RWSL will

start a new system track when the ASDE-X/DFW track number changes, causing loss of all safety logic information gathered up to that time.

The multiple track detection algorithms, when they determine such a situation, “fix” the problem so that the RWSL system will be none the wiser. In the first two cases, the algorithm changes the track numbers in the duplicate track’s reports to the number of the primary track, so that only a unique track number is seen by RWSL. In the third case, the algorithm creates a “false drop” message as explained in Section 2.2.2 below, so that the system track from the original track will be used when the second track is initiated, resulting in continuity in RWSL information.

2.2.2 Merging Split Mode S Tracks

Several instances of individual aircraft having multiple Mode S tracks simultaneously active in the surveillance data at DFW have been noted and analyzed. These cases appear to result from bad contacts in the avionics wiring harness that sets the Mode S address in the aircraft’s transponder. Logic has been incorporated into the surveillance improvement algorithms package to detect and repair these Mode S track splits. (The logic makes the assumption that the Mode S addresses of the split tracks will differ by at most one bit.)

The algorithm performs a search through the track file looking for a second track that is a “split” from the current track. The second track must be currently active and have been updated within the preceding 10 seconds. The position of the second track is extrapolated to the current time and the difference in horizontal position between the two tracks is calculated. The threshold for positional difference is 100 meters; if the positional difference between the two tracks exceeds the threshold, then the search continues with the next track in the track file.

Further code tests are performed with a second track that has passed the positional test. The number of bit differences between the Mode S codes of the two tracks is computed (assuming both tracks have Mode S codes). Mode S code “agreement” is defined as both tracks having a Mode S code and there being at most 1 bit difference between them. Similarly, the ATRBS Mode 3/A code agreement between the tracks is computed. (Note: if the input report lacks a validated Mode 3/A code, then the most-recent validated Mode 3/A code saved in the track file is used instead.) If the Mode S codes agree (allowing a single bit difference) and the ATRBS Mode 3/A codes agree, the tracks are considered to constitute a split situation.

Processing to determine which track number to maintain is performed as follows. If one of the two tracks has multilateration support and the other track does not, then the track with multilateration support is determined to be the real one. If both tracks have multilateration support, then the most mature Mode S track (based on the time of the first Mode S update for the given track) is determined to be the real one. Once the track number is selected, this number is used for all reports from either track. The non-selected

track has a cross-reference to the maintained track, and the maintained track has an inverse cross-reference to the non-selected track.

Some additional cross-referencing bookkeeping is required by the merging algorithm. First, the search for cross-reference tracks is performed from scratch for every report, in case the track situation has changed. Thus, the cross-reference is not remembered, and, more importantly, the cross-referenced track must have its inverse cross-reference field cleared. In addition, there may be cases where the track split is into three or more tracks – say A, B, and C. Then it is possible that A finds B and B finds C (the search order is random), so that A’s reports would use track number B while B’s reports would use track number C and a split would still exist. To prevent this behavior, one additional rule is enforced: if the cross-reference track itself has an existing cross-reference, that latter track number becomes the new cross-reference (i.e., in the example, the next time A finds B, A’s cross-reference becomes C, and only track C’s number is used for reports for all three tracks).

A problem could exist if the maintained track is the first to drop, as now the previously non-selected track must keep its track number for all its subsequent reports. This problem is overcome by the use of a “false drop” indication in the drop message for the maintained track. The “false drop” flag instructs the RWSL system not to drop the system track corresponding to that ASDE-X/DFW track. Thus the system track is available to fuse with the previously non-selected track when its number appears in reports, and system continuity is assured. The processing of drop messages to determine when to set the “false drop” flag is: set the flag whenever the track being dropped has an inverse cross-reference in its track file.

2.2.3 Merging Split ATCRBS Tracks

The input surveillance system is tasked with merging local tracks from different surveillance sources (Mode S multilateration, ATCRBS multilateration, one or more ASR-9’s) into a single system track. Often, this process will fail to work, and multiple tracks exist for the same aircraft. These tracks will have the same Mode 3/A code, so that finding them is possible. (Since Mode S code only exists in Mode S multilateration, only one of the tracks will have this code, and matching on Mode S code is not possible).

In a similar manner to the previous Mode S case, the algorithm performs a search through the track file looking for a second track that is a “split” from the current track. The time test and the position extrapolation are identical. The threshold for positional difference is 100 meters if both tracks have used multilateration in their surveillance. If either track has only ASR-9 surveillance (subject to greater positional errors), then the positional threshold is increased to 300 meters. The code test in this case is that the two tracks must have the same discrete (non-1200₈) ATCRBS Mode 3/A code. (Note: if the input report lacks a validated Mode 3/A code, then the most-recent validated Mode 3/A code saved in the track file is used instead.)

The processing to determine which track number to maintain is also performed as above, except that if neither or both tracks have multilateration data, so that maturity is used to decide, the track's first update time with the current Mode 3/A code is used (unlike Mode S, Mode 3/A codes can change for a track).

The cross-reference and inverse cross-reference handling is exactly the same as for Mode S (Section 2.2.2 above), as is the use of "false drop" flags to maintain system track continuity.

2.2.4 Handling Track Number Changes

For reasons not fully understood, the ASDE-X/DFW tracker will often change the track number used for an aircraft track (starting or termination of flight plan assignment may be one case). The sequence of events in such cases is use of one number, followed by a track drop message for that number, followed by use of the second number. Since the two tracks will not exist at the same time, the search mechanism detailed above will not find the duplication.

Luckily, in almost all cases, the track drop message is issued soon after the last report with the original track number. This time interval is much shorter than the usual 8 seconds or longer coast period before a track is normally dropped. Thus, the rules for identifying track number changes are as follows:

1. time between a track drop message and the last track update must be less than 5 seconds (premature drop), and
2. number of track updates must be at least 10 (non-established tracks often drop early)

When this situation is found, the track drop is converted to a "false drop." As explained above, the "false drop" flag instructs the RWSL system not to drop the system track corresponding to that ASDE-X/DFW track. Thus the system track is available to fuse with the new track when it is seen, and information continuity is maintained.

2.3 IDENTIFICATION OF INVALID MODE C ALTITUDES

The logic in this section of the surveillance improvement algorithms package detects cases of erroneous reported altitudes in the input reports. This logic goes beyond the altitude validity flag already contained in the input target report (based upon the sensor's altitude detection status flag bits) using track history information contained in the track file.

The altitude validity algorithm assumes a maximum valid altitude error of 1000 feet plus a maximum altitude rate of 100 feet per second. The algorithm computes the total allowable altitude error (termed "delta") since the last valid altitude update to the track. (Note: the first altitude update on a given track cannot be validated, but merely initiates the process.) Only tracked input reports with altitudes

declared valid by the input sensor are used in the altitude validity algorithm. ADS-B surveillance reports are not used in this algorithm.

Three altitude values are maintained in the track file for each track entry. One value holds the most-recently validated altitude, while the second holds the previous validated altitude. If the magnitude of the difference between the report altitude and the most-recently validated altitude for this track is within “delta” (the usual case), then the altitude is declared ‘valid’. Otherwise, if the magnitude of the difference between the report altitude and the previous validated altitude for this track is within “delta” (indicating the last altitude for the track may have been spurious), then the altitude is declared ‘valid.’ If neither test is satisfied, the report altitude is declared invalid.

2.4 REJECTION OF INVALID TRACK VELOCITIES

The ASDE-X/DFW fusion tracker has been seen to generate significant transients in its ground speed estimates when tracks are first initialized. Logic in the surveillance improvement algorithms package flags these fluctuations so that the subsequent RWSL processing will not depend on the possibly spurious tracker velocity estimates for these tracks.

The surveillance improvement algorithms package maintains the current ground speed estimate for each track entry. (Note: the first report on each track is flagged as invalid because the track ground speed must be initialized.) For subsequent updates of the track, until its velocity has been declared valid, the following tests are performed. If the change in the track’s ground speed since the previous update is greater than 30 knots, and the change is at least 30 percent of the track’s previous ground speed, then the track’s velocity is declared invalid. Requiring at least a 30 knots absolute change and a 30 percent relative ground speed change avoids issues with the measurement of very slow (taxi) speeds and the fact that surveillance measurement errors can result in apparent ground speed fluctuations of this order. Once a track’s velocity is declared valid, all further reports are declared valid. Future sudden changes in velocity are handled by the track jump logic of Section 2.1.4 above.

2.5 REJECTION OF ‘0607’ SPURIOUS TRACKS

Rockwell-Collins TPR-901 transponders manufactured between March 2004 and December 2005 have a known operational anomaly where the transponder may occasionally respond with an ATCRBS Mode 3/A code of 0607₈ instead of the proper Mode 3/A code set by the pilot. These anomalous reports are caused by a timing problem in the transponder logic dealing with the reception of a Mode S interrogation while in the process of outputting an ATCRBS reply. Before the anomaly, Mode 3/A replies have the proper code; after the anomaly, they have code 0607₈. The Mode C altitude replies are unaffected, and all have the same code during the reply run. Thus, the usual result of the anomaly is that two ATCRBS reports will be generated by the processing code—the first with the proper Mode 3/A and Mode C codes, the latter with Mode 3/A code 0607₈ and no Mode C replies. The anomalous split reports can be frequent enough to initiate secondary tracks on aircraft equipped with these transponders. On

occasion, the position of the split track may be in error, leading to a false track situation. Unfortunately, Mode 3/A code 0607₈ is a legal code, so reports with this code cannot simply be rejected. Although this transponder defect was slated for repair on all domestically registered aircraft in 2006, rejection of the spurious tracks was undertaken to eliminate the possibility of surveillance corruption due to noncompliant transponders.

The logic in the surveillance improvement algorithms package deals with 0607₈ ATCRBS split tracks by counting the number of times that a track with the 0607₈ Mode 3/A code has seen an update with Mode C altitude data. (As described above, it is assumed that the spurious reports will lack altitude data.) The counting logic omits data from multilateration surveillance (since multilateration ATCRBS reports are generated from a single reply). If the track has received at least 3 consecutive reports having Mode 3/A code 0607₈ and Mode C altitude data, then the 0607₈ report code is considered valid. If the track has not received at least 3 consecutive reports having Mode 3/A code 0607₈ and Mode C altitude data, and the current input report for the track lacks Mode C altitude data, then the track's 0607₈ code counter is reset. If the track with Mode 3/A code 0607₈ has not had multilateration support and has not seen at least 3 consecutive reports with Mode C altitude support, the new input report is flagged "false" so that it will not be used by subsequent RWSL processing.

3. SAMPLE RESULTS

This section presents example results illustrating the application of various algorithms included in the ASDE-X/DFW surveillance improvement algorithms package. In particular, several actual cases where REL or THL lights were incorrectly activated are included. All data employed was collected from the multisensor ASDE-X surveillance system installed at the DFW airport.

As discussed in Section 2.1.5, the RU scoring method employed by the false target algorithm is dependent upon the number of coverage samples in the set of nine grid cells centered at the target position; the most reliable method requires that at least 500 samples exist in this set. Figure 1 presents a map of the Mode S statistical samples in each grid cell on the DFW airport: red cells have 500 or more samples in the cell by itself, while blue cells meet this threshold by including the surrounding cell set. As seen, all runways and major taxiways are covered.

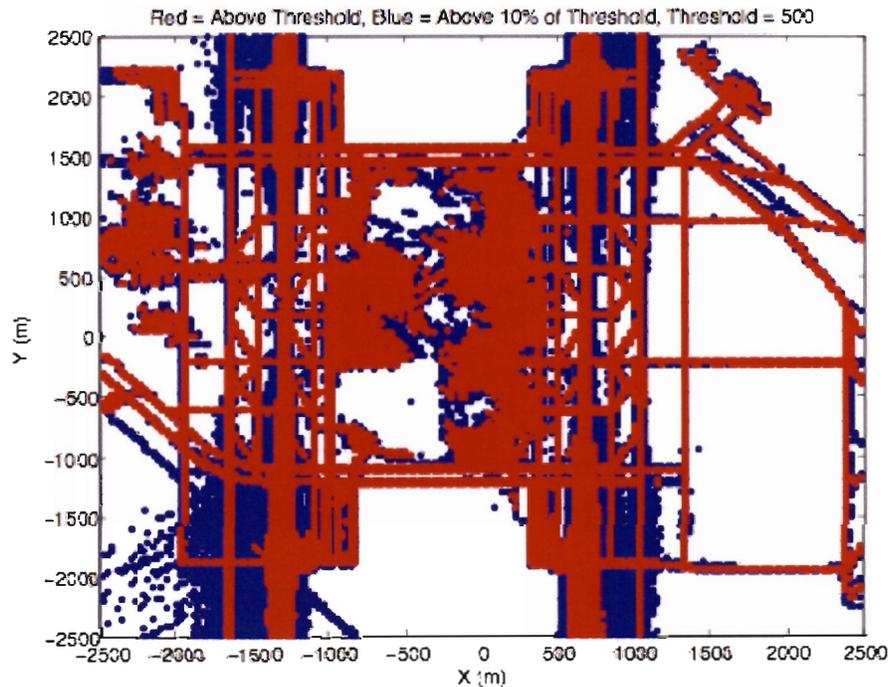


Figure 1. RU Mode S grid cell coverage for DFW.

The false target algorithm was tested on several complete days of data recorded at DFW. A sample day's false target results are presented in Figure 2. The sample day had approximately 4,000,000 total target reports; approximately 200,000 (5%) of these target reports were determined to be false. As seen, although most false targets were in the gate areas, numerous false targets exist in the critical safety logic regions. Thus, the potential for false light activations existed, and in fact several such events occurred on the day in question.

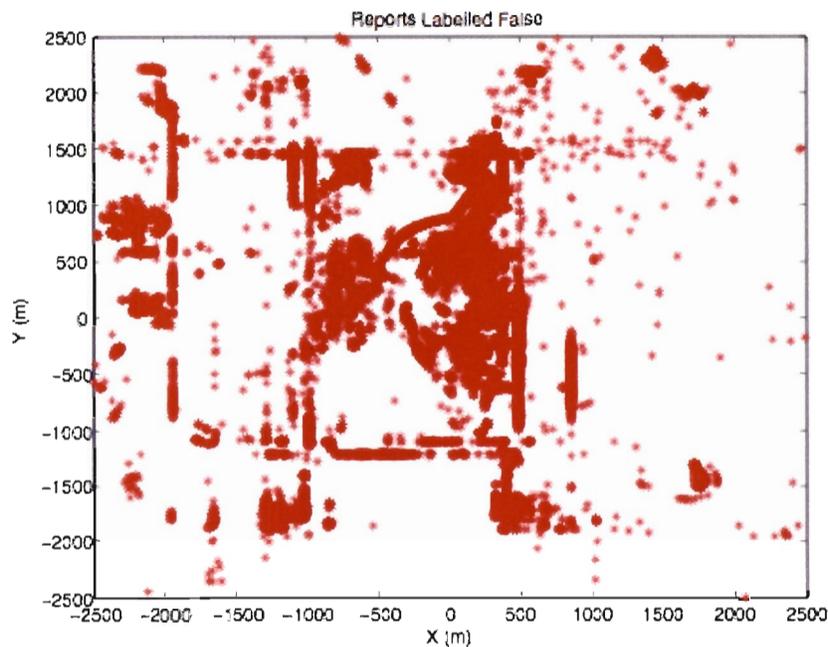


Figure 2. Targets declared false (24 hours) at DFW.

One of the reasons the ASDE-X/DFW tracker occasionally creates false tracks is that the multilateration report stream it has to work with may include many reports with false positions. Figure 3a presents the multilateration track for a single Mode S address. As can be seen, the data has considerable dispersion. Thus, it is not surprising (as indicated by Figure 3b) that the ASDE-X/DFW tracker produced numerous false tracks in addition to the real track. One of these false tracks was located on the runway and activated a false REL light. The algorithm described in Section 2.1 successfully identifies and suppresses all these false tracks.

**Multilateration Track 704
Mode S Address A64C88
5 Minutes of Data**

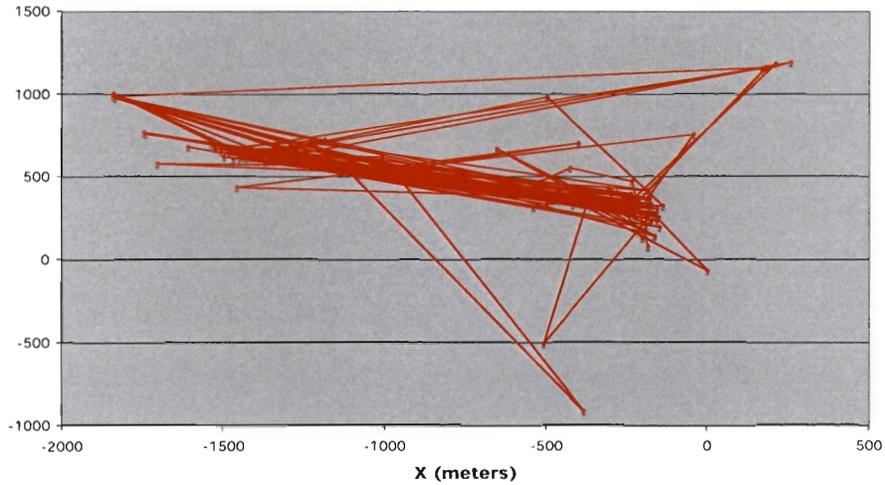


Figure 3a. Multilateration reports received for Mode S address A64C88.

**ASDE-X Tracks
Mode S Address A64C88**

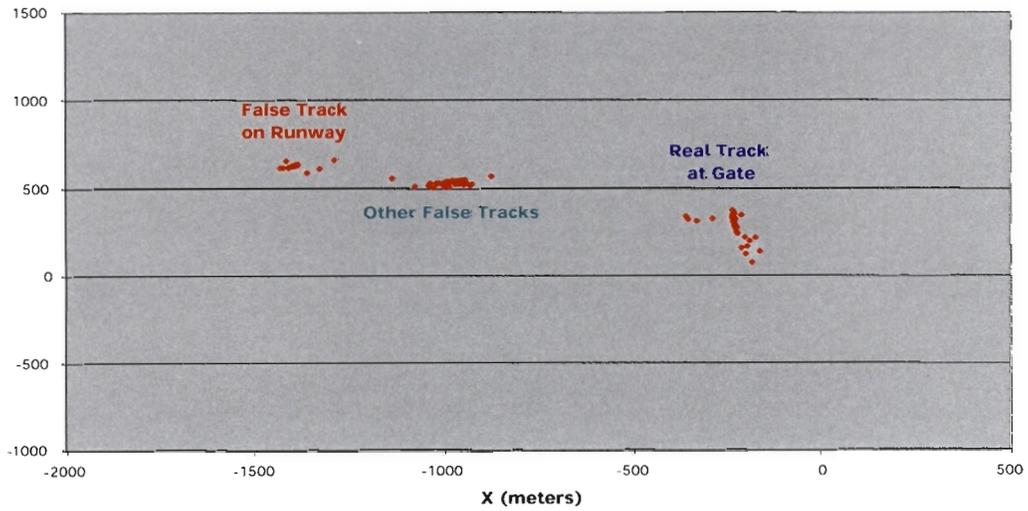


Figure 3b. Tracks (real and false) declared by ASDE-X/DFW from above reports.

In addition to creating false tracks, the ASDE-X/DFW tracker will, on occasion, switch a track from a true position to a false position when reports at the true location cease and reports start to be produced at the false position (often caused by the initiation of multipath when the aircraft moves to a location where a building is illuminated by the return signal). Figure 4a illustrates such a situation, where the new position resides on the runway, activating the REL lights. The viewer will not realize this jump has occurred, as the ASDE-X/DFW tracker smoothing function produces output as shown in Figure 4b, creating the impression the aircraft has taxied to the runway. The jump logic of Section 2.1.4 successfully identifies this case and suppresses the reports on the runway.

**Example of Track Jump
Track 508 Declared False After Jump**

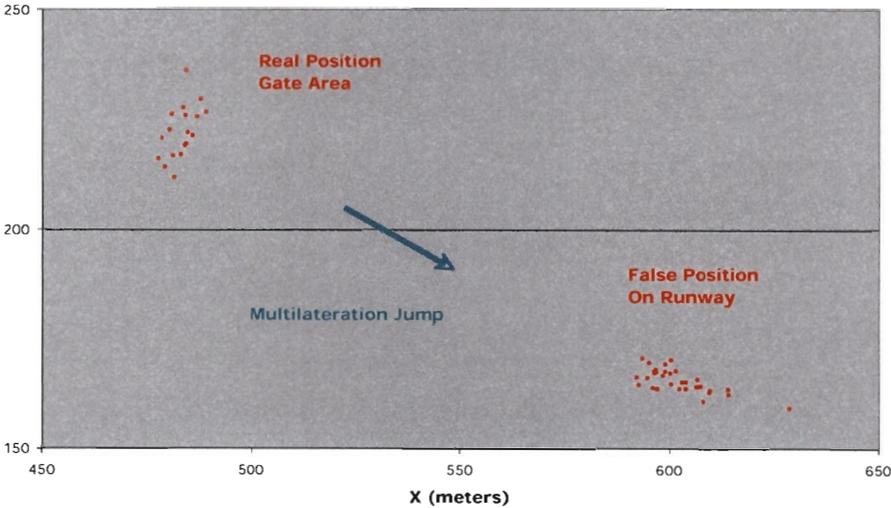


Figure 4a. Example of ASDE-X/DFW track jump, reports input to tracker.

ASDE-X Fusion Tracker Output Jump Smoothed

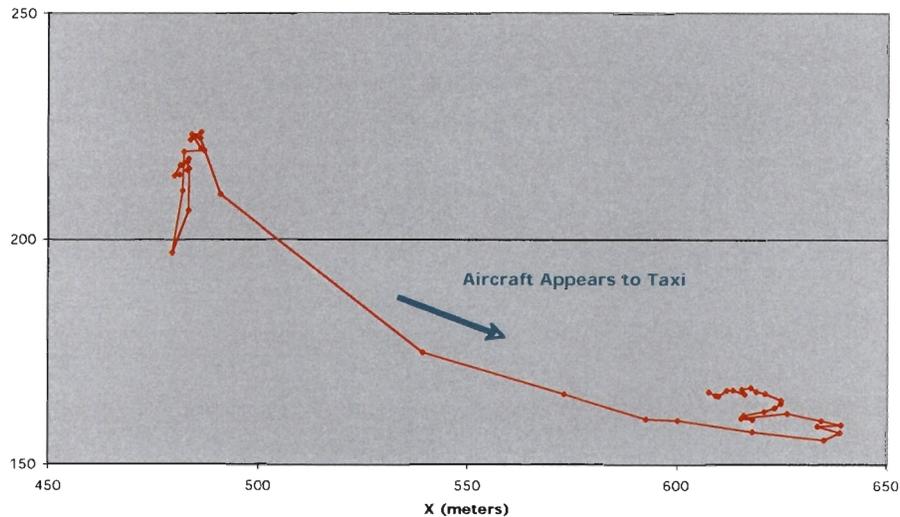


Figure 4b. Example of ASDE-X/DFW track jump, positions output by tracker.

The ASDE-X/DFW tracker often creates two tracks for the same aircraft, causing the RWSL safety logic to believe two aircraft exist. When the split situation ceases, and one of the tracks no longer receives updates, the “second” aircraft is assumed to have stopped at its last reported position until the track times out. If this position is in a sensitive region (such as on a runway), the lights will operate incorrectly. Figures 5a and 5b illustrate two cases where the REL and THL lights remained on after the real aircraft cleared the activation zones. The first case was due to a faulty transponder emitting 2 different Mode S addresses, while the second case was caused by the ASDE-X/DFW tracker failing to fuse multilateration and ASR-9 report streams. The algorithms in Sections 2.2.2 and 2.2.3, respectively, correctly resolve these cases by outputting all reports in each case with a single track number.

False light activations can occur if the tracker velocity is output with too large a value, such that an aircraft safely stopped (or moving slowly) is projected onto the runway. These invalid velocities typically occur when a track is initiated and sufficient data for accurate smoothing is not yet available. The algorithm in Section 2.4 attempts to identify and label these invalid velocities by checking for inconsistent early velocity estimates. Figure 6 presents a scatter plot of all velocity estimates output by the ASDE-X/DFW tracker, as a function of track update number, for all aircraft whose actual velocity is 10 knots or less. As can be seen, the dispersion is quite large at first, with errors decreasing over time. The red dots indicate velocities labeled as invalid by the algorithm. Unlabelled bad points are due to consistent tracker velocity errors, which are assumed to indicate faster moving aircraft.

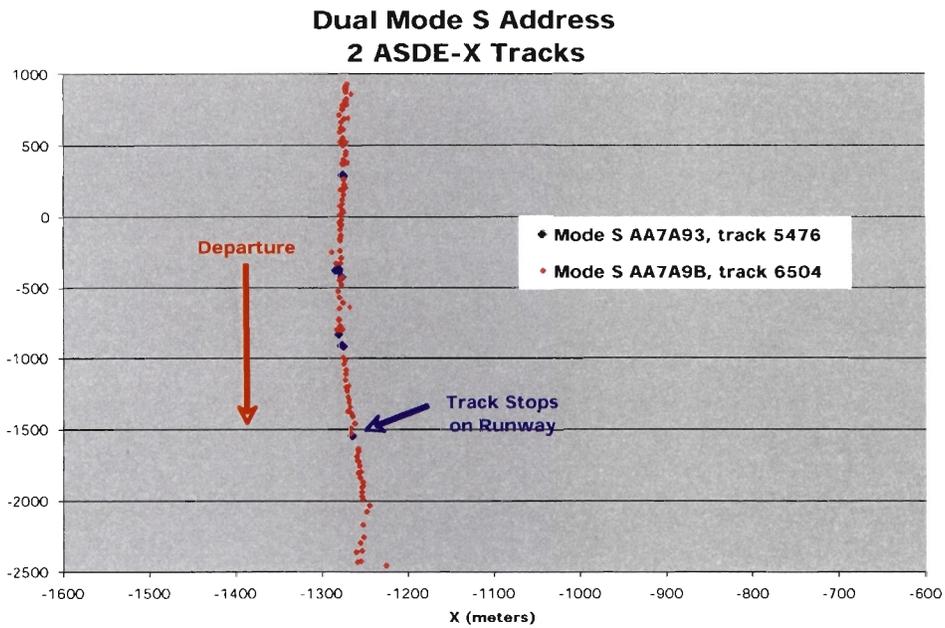


Figure 5a. Example of ASDE-X/DFW track split due to Mode S Transponder failure.

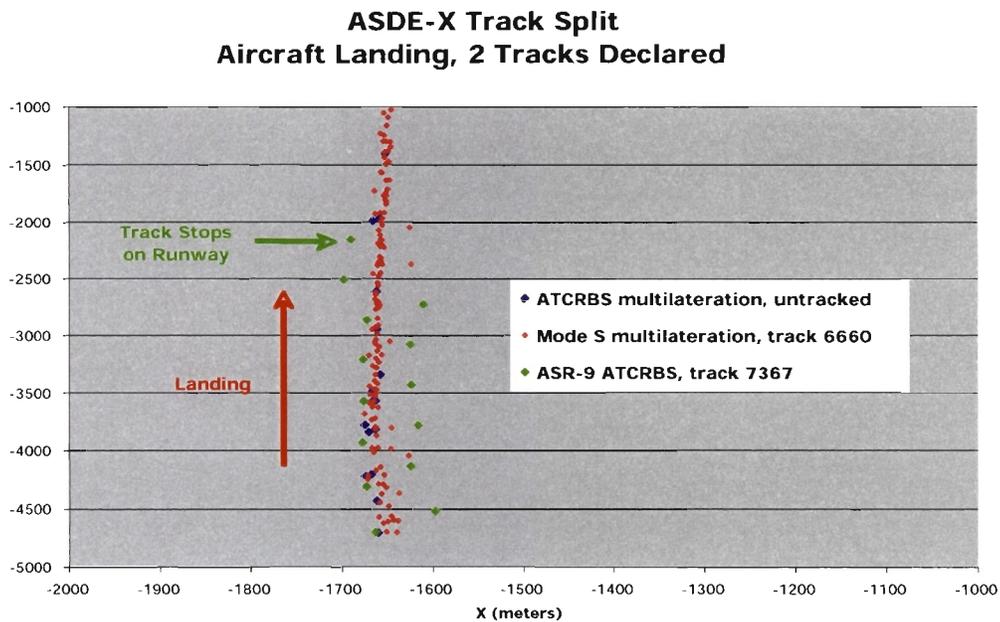


Figure 5b. Example of ASDE-X/DFW track split due to ASDE-X/DFW Fusion failure.

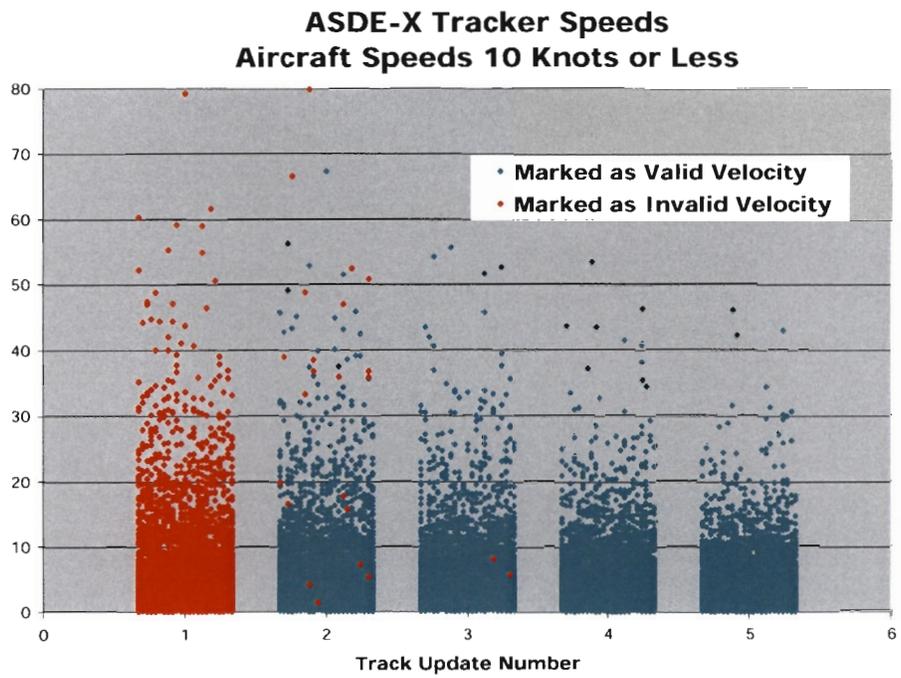


Figure 6. Invalid velocity estimates for slow moving aircraft.

4. SUMMARY

This report has described and documented the set of ASDE-X surveillance improvement algorithms developed for the ASDE-X system under evaluation at DFW as part of the RWSL project. These algorithms have been demonstrated to reduce the incidence of false targets appearing on runways and taxiways on the RWSL display significantly—there have been no such false-target cases observed in live testing since these algorithms were incorporated. They have also eliminated spurious altitudes and speeds that have plagued the RWSL safety logic. Finally, they have increased significantly the probability that all reports from a given aircraft will be output with a unique track number.

ACRONYMS

ADS	Automatic Dependent Surveillance
ADS-B	Automatic Dependent Surveillance-Broadcast
ASDE	Airport Surface Detection Equipment
ASDE-X	Airport Surface Detection Equipment – Model X
ASR	Airport Surveillance Radar
ATC	Air Traffic Control
ATCRBS	Air Traffic Control Radar Beacon System
DFW	Dallas/Fort Worth Airport
IP	Internet Protocol
MDT	Maintenance Display Terminal
MDT-TP	Maintenance Display Terminal – Target Processor
MSDP	Multi-sensor Data Processor
REL	Runway Entrance Light
RU	Receiver Units
RWSL	Runway Status Lights
THL	Takeoff Hold Light