NASA/TM—2014–216520

A User Guide for Smoothing Air Traffic Radar Data

Ralph E. Bach
Aerospace Computing, Inc.

Russell A. Paielli
NASA Ames Research Center

May 2014

A User Guide for Smoothing Air Traffic Radar Data

Ralph E. Bach
Aerospace Computing, Inc.

Russell A. Paielli
NASA Ames Research Center

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California, 94035-1000

May 2014

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

This report is also available in electronic form at

http://www.aviationsystemsdivision.arc.nasa.gov/publications/index.shtml

Abstract

Matlab software was written to provide smoothing of radar tracking data to simulate ADS-B (Automatic Dependent Surveillance-Broadcast) data in order to test a tactical conflict probe. The probe, called TSAFE (Tactical Separation-Assured Flight Environment), is designed to handle air-traffic conflicts left undetected or unresolved when loss-of-separation is predicted to occur within approximately two minutes. The data stream that is down-linked from an aircraft equipped with an ADS-B system would include accurate GPS-derived position and velocity information at sample rates of 1 Hz. Nation-wide ADS-B equipage (mandated by 2020) should improve surveillance accuracy and TSAFE performance. Currently, position data are provided by Center radar (nominal 12-sec samples) and Terminal radar (nominal 4.8-sec samples). Aircraft ground speed and ground track are estimated using real-time filtering, causing lags up to 60 sec, compromising performance of a tactical resolution tool. Offline smoothing of radar data reduces wild-point errors, provides a sample rate as high as 1 Hz, and yields more accurate and lag-free estimates of ground speed, ground track, and climb rate. Until full ADS-B implementation is available, smoothed radar data should provide reasonable track estimates for testing TSAFE in an ADS-B-like environment. An example illustrates the smoothing of radar data and shows a comparison of smoothed-radar and ADS-B tracking. This document is intended to serve as a guide for using the smoothing software.

This page is intentionally left blank.

I Introduction

This document describes a Matlab[1] program to smooth radar position data from flights in the National Airspace System (NAS). The data come from Air Route Traffic Control Center (ARTCC) and Terminal Radar Approach Control (TRACON) sources. The TRACON source is given priority over the ARTCC source: where there is overlap (usually below 20,000 ft), TRACON tracks are used. Each Center track record contains nominal 12-sec samples; each Terminal record has nominal 4.8-sec samples. Both record sources contain dropouts and wild points, although TRACON tracks have less uncertainty. Each track record comprises time-tagged values of aircraft position (x, y in stereographic coordinates), Mode-C altitude, and real-time filter estimates of ground speed, ground track, and climb rate. The filter estimates are error-prone and can have a lag of 60 sec.

Smoothing of the track records reduces wild-point errors and interpolates through data dropouts, and because it is an off-line process, it provides lag-free estimates of ground speed and ground track. Smoothed tracks can simulate the data stream down-linked from aircraft equipped with an Automatic Dependent Surveillance-Broadcast (ADS-B) system. The ADS-B stream includes GPS-derived position, ground speed, ground track, and Mode-S altitude information. The advantage of GPS tracking is that it provides position accuracy at least an order of magnitude better than ARTCC radar tracking, and with a sample rate of 1 Hz, accurate and nearly lag-free estimates of ground speed and ground track. Currently, less than 20% of commercial aircraft are ADS-B equipped, and full implementation is not mandated until 2020.[2]

The smoothing code was written to support testing of TSAFE[3-5] (Tactical Separation-Assured Flight Environment) in a simulated ADS-B environment. TSAFE is intended to handle air-traffic conflicts left unresolved when loss-of-separation is predicted to occur within approximately two minutes. Radar-derived track records are an essential part of each standard TSAFE input file.[6] When those tracks become available from an ADS-B source, it is expected that TSAFE conflict prediction and resolution will be improved. Until full ADS-B implementation is available, however, smoothing radar data should provide reasonably accurate and lag-free track estimates for testing TSAFE in an ADS-B-like environment. Recently, access to a 42-day (03/04/2013 – 04/16/2013) data base consisting of NAS-wide aircraft flight and ground traffic data was made available through the NASA Data Warehouse.[7] These data include some from flights equipped to downlink ADS-B tracks. This made it possible to find flights in the data base that could be used to compare results of smoothed-radar and GPS-derived tracks.

There are two versions of the smoothing code: one is a batch version called *batchSmooth*, used to smooth all track records in a data file (there may be several thousand flights included). A second version provides printed output including plots, and is used for code testing or when a detailed analysis of a single flight is required. This version, called *smoothTracks*, can also be used in batch mode. The basic algorithms are common to both versions. The algorithms are outlined next in Section II. More information on the core smoothing algorithms is covered in Appendix B. Section III describes how to use the program and Section IV presents an example of the smoothing procedure for a single flight. This flight was included in the ADS-B data base, and was used to compare radar position smoothing and ADS-B tracking shown in Section V.

II Algorithms

The radar tracking data sent to TSAFE is derived from Center radar (nominal 12-sec update) and Terminal radar (nominal 4.8-sec update). The track data consist of North and East position records (in a stereographic coordinate system), and pressure altitude (from an onboard transponder). The position and altitude records are non-periodic because the radar "hit" rate varies from the nominal. In addition, there will be data dropouts and gaps where wild points have been removed. The basic tasks performed by the software described in this document are smoothing, wild-point removal and interpolation to a common time base, and smoothing of the interpolated records. The tasks are summarized in this section, starting with a review of the smoothing algorithm.

Smoothing

Here smoothing is done by sliding a cubic polynomial, fit to an odd number of consecutive data points (*narc*), through a record *zR* of length *npts*, taking each "interior" center point as the smoothed estimate.[8] There are a number of ways to handle boundary requirements. Perhaps the simplest is to use as boundary-segment estimates the first *(narc+1)/2* points of the first narc-point cubic fit to *zR*, and the last *(narc+1)/2* points of the last *narc*-point fit to *zR*. Then, smoothed estimates of the interior portion from point *(narc+1)/2* to point *npts-(narc+1)/2* are obtained, point by point, by sliding the cubic arc from starting point 2 to ending point *npts – 1*, taking the center point of each fit to complete the smoothed record. The procedure is illustrated in Figure 1 by a moving arc fit with *narc* = 5 points. For the interior points, only the center point of each fit contributes to the smoothed record.

Wild-Point Removal and Interpolation

Since the smoothing program has been designed for use in "batch" mode, a simple statistical test is used to identify wild points. A smoothing pass is made on each raw position record, and the mean and standard deviation computed for the error sequence (defined as the difference between the smoothed and raw records at each time point). If a point in the error sequence is outside of three standard deviations, it is considered a wild point, and the corresponding point in the raw record is removed.

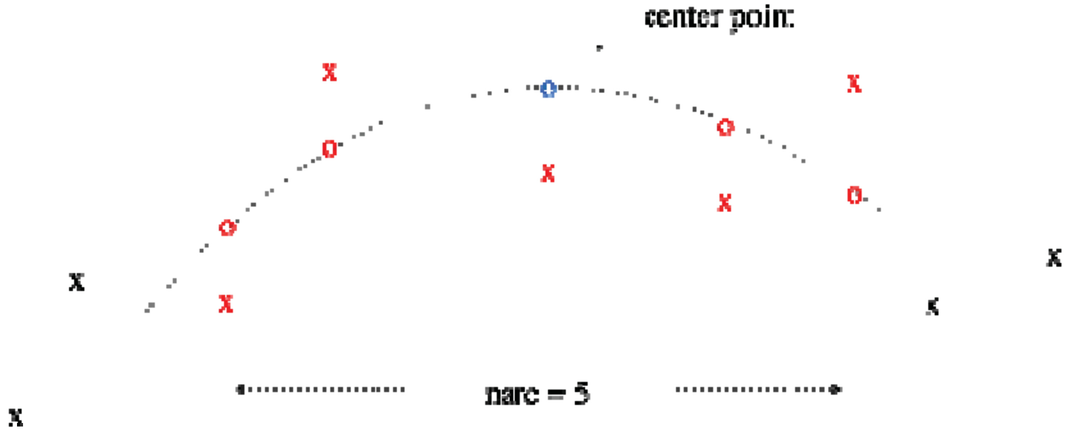

Figure 1. Moving-arc least-square fit to 5 points: x = raw; o = fit.

Following wild-point removal in position and altitude records, another smoothing pass is made on each (non-periodic) record. The smoothed representation is used to compute periodic samples at exactly 1, 2, 3, 4, 6, or 12 sec. The period is chosen by the user (the default value is 6 sec).

The default number of arc points *narcD* for raw-data smoothing has been chosen to reasonably capture the aircraft dynamics from a nominal 12-sec Center radar “sweep” rate. Since terminal radar data (with a 4.8-sec sweep) are included in the track records, a default arc length will be replaced by

narcR = (nearest odd integer) > (12/*avgPer*)**narcD*

where *narcR* is the number of arc points used to smooth a raw record, and *avgPer* is the average period of the raw record.

Smoothing of Interpolated Records

Two smoothing passes are made on each interpolated record. Following this, the number of arc points is adjusted again, to the value

narcI = (nearest odd integer) > (*avgPer*/*hFin*)**narcR*

where *narcI* is the number of arc points used to smooth an interpolated record and *hFin* is the user-chosen sample period.

After processing of all records, the climb rate, ground speed, and ground track are computed from time-derivative estimates provided by the smoothing algorithm. In the version *smoothTracks*, there is also a “single-run” mode, which has screen output and plots to show how the smoothing process works and to facilitate debugging.

III Using the Smoothing Program

To set up the smoothing program, in your home directory create a directory called "matlab" containing a file called "startup.m" that contains one line:

```
addpath path_to_smoothing_algorithm_directory
```

All the smoothing algorithms should reside in this directory.[9] In the directory chosen to run Matlab there should be a TSAFE input file called *TSAFE.in.unsmoothed* (see section IV). To run *smoothTracks* (or *batchSmooth*) from this directory give the command

```
matlab –nosplash –nodesktop -r smoothTracks (or batchSmooth)
```

to smooth the track data. In batch mode, a new file called *TSAFE.in.smoothed* will be created which will be the same as *TSAFE.in.unsmoothed*, except that smoothed tracks will replace the original raw track records. *Note*: All routines mentioned in this section and the next are described in Appendix B.

Running smoothTracks

This version of the smoothing code is interactive. It can provide screen output, including plots that are useful for code testing or when a detailed analysis of a flight is required. It also can be used in batch mode. When the Matlab call to run *smoothTracks* is given, the following screen output appears:

```
The TSAFE input file to be read is TSAFE.in.unsmoothed
```

Final sample period: 1, 2, 3, 4, 6, 12 sec (default 6):

Points for altitude smoothing: -- 5, 7, 9, -- (default 5):

Points for X-Y smoothing: -- 9, 11, 13, -- (default 9):

Minimum flight level for inclusion in record (default 120):

Parameters: hFin = 6*; nmh* = 2*; nmp* = 4*; minAlt* = 12000 ft
Note: nmh and nmp may increase if TRACON data included

If other than a default value is desired, respond to the prompt by entering the value. Recall that the default arc points for altitude (h) and plan-view (x-y) smoothing have been chosen to match the nominal 12-sec Center radar "sweep" rate. *Note*: A smoothing parameter *nm* is related to the arc length *narc* by *nm = (narc –1)/2*.

Next, the routine *getACIDList* determines a list of unique aircraft call signs by parsing the input file for lines beginning with "FLT." These flight records correspond to the filed flight plans and provide basic information about the flight, such as cruise altitude and airspeed, aircraft type, and equipage. After the list has been created, the screen output is, for example,

There are 5648 *call signs contained in the flight records*

Note: Not all unique call signs will have track records that are "smoothable." Some aircraft may never have departed; some may never rise above *minAlt* (only records with altitude above that level are included), or the record length may contain fewer points than the longest arc.

Now the user can choose the flights whose tracks are to be smoothed. Usually the choice would be either one or all, although any number from one to all may be chosen. If and only if one flight is chosen, extended screen output, including plots, is displayed. This can help answer questions about a solution or test a code modification.

The user will be given a series of prompts. As an example, consider the following sequence of prompts, each followed by a typical user response for purposes of illustration:

For a single call sign, enter it within single quotes: 'AAL1225/KDFW'

For a single run there will be a prompt to have plots printed:

Enter one to have plots printed: 1

The complete output for a single smoothing run is presented in a later section.

On the other hand, if it desired to select a range of flights and to store results in *TSAFE.in.smoothed*, the prompts will be

Enter indices for a range of smoothing runs [m1 m2]: [1 5648]

Enter one to store smoothing results: 1

Here the screen output will be

The smoothed tracks will be stored in TSAFE.in.smoothed

In this example all runs have been selected for smoothing and the results are to be stored. Since the number of flights chosen is greater than 1, the only screen output for each smoothing run is, for example,

```
Working on run: 127 AAL1225/KDFW
```

After all processing has been completed, a screen message will appear that indicates what fraction of the track records have been smoothed:

```
Flight track records that have been smoothed: 4194/5648
```

Note: Not every track record could be smoothed. In this case, records for 4194 call signs will have been smoothed. Finally, following the creation of file *TSAFE.in.smoothed*, this message will appear:

```
Smoothed track records that have been stored: 4194
```

Running batchSmooth

This batch version of the smoother is not interactive. The default parameter values are used. (For other than default values, *smoothTracks* can be used in batch mode.) All flight track data will be smoothed and stored in the file *TSAFE.in.smoothed*. Once the call to Matlab to run *batchSmooth* is given, the only screen output (for the same input file used in the last example) is

```
running batchSmooth

input file: TSAFE.in.unsmoothed

output file: TSAFE.in.smoothed

Parameters: hFin = 6; nmh = 2; nmp = 4; minAlt = 12000 ft

There are 5648 call signs contained in the flight records

Flight track records that have been smoothed: 4194/5648

Smoothed track records that have been stored: 4194
```

IV A Smoothing Example

With access to a data base consisting of NAS-wide air-traffic data that includes some down-linked ADS-B tracking data, it became feasible to find flights that had both radar and ADS-B track sources. A search was made for such flights in the ZFW Center during a four-hour period on the day 03/24/2013. In that period, there were a total of 1480 flights; only 106 of these flights were equipped with ADS-B. One of those flights, AAL1847, is presented in this section as an example of smoothing radar track data. The track source was the ZFW ARTCC and TRACON. A comparison of the smoothed-radar and GPS-derived tracks is shown in the next section.

The example presented here is a single smoothing run interaction with the user, with screen output, including plots. It was obtained by calling Matlab with the command

```
matlab -r smoothTracks
```

The program responds with:

```
The TSAFE input file to be read is: TSAFE.in.unsmoothed

Define the smoothing parameters:

Final sample period: 1, 2, 3, 4, 6, 12 sec (default 6):

Points for altitude smoothing: -- 5, 7, 9 -- (default 5):

Points for X-Y smoothing: -- 9, 11, 13, -- (default 9):

    Minimum flight level for record inclusion (default 120):

Parameters: hFin =  6; nmh = 2; nmp = 4; minAlt = 12000 ft
Note: nmh and nmp may increase if TRACON data in records

There are 5272 call signs contained in the flight records

For a single call sign, enter it within single quotes: 'AAL1847/KBOS'

The index for the call sign is 429

Enter one to have plots printed: 1
```

In this example the user has chosen to use the default values for the sample period, smoothing parameters, and minimum altitude. The choice of a single call sign (AAL1847/KBOS) causes routine *getRawTRKData* to read the track records for that aircraft and generate the following screen output:

```
Working on run 429: AAL1847/KBOS

All track data read: avPer =  9.4 sec; npts = 177
 first pt: time(1) = -0.3; last pt: time(177) = 1661.9
 (37 track points discarded less than 12000 ft)
 After processing, add 1364131857 sec to the time record

Found 2 track segments (C = Center; T = Tracon):

 C: 113 pts, from pt   1 to pt 113 (-0.3 - 1355.5 sec)
 T:  64 pts, from pt 114 to pt 177 (1365.7 - 1661.9 sec)

 Average period for all Center data: 12.1 sec
 Average period for all Tracon data:  4.7 sec

 Percentage of track data from the Tracon: 36.2

Altitude cruise patches found:

  1 patch(es) at altitude 36000
  1 patch(es) at altitude 35000

 No 100 ft jump corrections in cruise

WP Smoothing parameters: nmh = 3; nmp = 5
```

The routine *removeAltJumps* analyzes cruise portions by determining patches of constant altitude and looking for discrete “jumps” in altitude before and after each patch. A patch is defined to have a minimum of 7 consecutive points at the same altitude. This procedure is useful for removing artifacts in the smoothed record. For this flight, no jumps were found in the Mode-C record.

Next, the routine *procTRKData* performs wildpointing, interpolation, and smoothing of altitude, x (East), and y (North) tracks for each aircraft's raw radar data record. Because of terminal track hits (at 4.8 sec) during the last part of descent from cruise, the average period is 9.4 sec. The smoothing parameters for wildpointing will be *nmh = 3* and *nmp = 5*. The following output shows wild-pointing results provided by routines *findStatWP_H* and *findStatWP_XY*:

```
Find wild points, interpolate to 6 sec samples:

 The fit to hR on pass 1:
   mean =  0.309, stdv = 33.417

 3 WPs have been removed from zR on pass 1

 The fit to h1 on pass 2:
   mean =  0.220, stdv = 29.407

 3 WPs have been removed from z1 on pass 2
 The fit to xR on pass 1:
   mean =  0.000, stdv =  0.108

 1 WPs have been removed from xR, yR on pass 1

 The fit to y1 on pass 2:
   mean = -0.000, stdv =  0.145

 3 WPs have been removed from x1, y1 on pass 2

 The fit to x2 on pass 3:
   mean = -0.000, stdv =  0.103

 1 WPs have been removed from x2, y2 on pass 3

 The fit to y3 on pass 4:
   mean =  0.001, stdv =  0.108

 1 WPs have been removed from x3, y3 on pass 4

Final smoothing parameters: nmh = 3; nmp = 6
```

After wildpointing a record, the routine *interpRec* interpolates to 6-sec samples, and the smoothing parameters are adjusted once again, to *nmh = 3* and *nmp = 6*. Next, two smoothing passes are made (each by the routine *smoothPass*, which is called by the routine *smoothTRKRec*) on the altitude and planview (x, y) records. This produces the following screen output:

```
Working on corrected AAL1847/KBOS Alt record:

Smoothing Pass 1:

 timI(1) = -0.0;  timI(278) = 1662.0

Fit on pass 1 of AAL1847/KBOS Alt record:
  mean =  0.052, stdv =  5.378

Smoothing Pass 2:

 timI(1) = -0.0; timI(278) = 1662.0

Fit on pass 2 of AAL1847/KBOS Alt record:
  mean =  0.036, stdv =  1.889

Working on corrected AAL1847/KBOS X record:
```

```
Smoothing Pass 1:

  timI(1) = -0.0;  timI(278) = 1662.0

Fit on pass 1 of AAL1847/KBOS X record:
  mean =  0.000, stdv =  0.015

Smoothing Pass 2:

 timI(1) = -0.0; timI(278) = 1662.0

Fit on pass 2 of AAL1847/KBOS X record:
  mean =  0.000, stdv =  0.004

Working on corrected AAL1847/KBOS Y record:

Smoothing Pass 1:

  timI(1) = -0.0;  timI(278) = 1662.0

Fit on pass 1 of AAL1847/KBOS Y record:
  mean = -0.000, stdv =  0.016

Smoothing Pass 2:

 timI(1) = -0.0; timI(278) = 1662.0

Fit on pass 2 of AAL1847/KBOS Y record:
  mean = -0.000, stdv =  0.005
```

After smoothing, the altitude record is discretized to 25-ft intervals to simulate Mode-S altitude samples down-linked from aircraft equipped with an ADS-B system. Finally, the position-derivative estimates provided by the routine *smoothPass* are used by the routine *GetRateEstimates* to estimate the records of ground speed, ground track, and climb rate. The final screen output, provided by *procTRKData* is

```
Position derivatives were used to compute estimates
of ground speed, ground track, and climb rate
```

Raw radar and smoothed x-y time histories are shown on a planview display in Figure 2. (*Note*: From this point on, the figures cited in this paper will all be found in Appendix A). The origin indicates that the aircraft is flying in a southwesterly direction. Raw and smoothed Mode-C altitudes are shown in Figure 3. Here the increased data density below 18000 ft indicates that the raw tracks are from terminal radar. For the velocity plots of Figures 4 and 5 (ground speed and ground track), back-differenced raw x-y values are shown with the corresponding smoothed estimates. Back differencing yields the least delay among real-time filters (and the most "noisy" derivative estimates). Notice that the velocities are less noisy during the terminal-radar portion of the altitude tracks.

V Comparison of Smoothed and ADS-B Tracks

The ADS-B–equipped flight AAL1847/KBOS was found within the time window of the data base available through the NASA Data Warehouse. Each ADS-B track consists of GPS position, GPS-derived ground speed and ground track, and Mode-S altitude. Although the downlink rate is nominally 1 Hz, the data base contains only every 5th track sample, which is still adequate to compare GPS ground speed and ground track with the smoothed radar estimates. The AAL1847 ADS-B tracks were accessed and a common time interval was determined for radar and ABS-B track records. The code written to perform these tasks is not included in the smoothing "Core Routines" described in the Appendix.

The screen output of the GPS-derived track acquisition is:

```
  ADS-B track data acquired: npts = 754

  The ADS-B data start time is 1364126400

  0 groundspeed wild points detected
  0 groundtrack wild points detected

  Check Mode-C altitude:

  Altitude cruise patches found:

   16 patch(es) at altitude 36000
    1 patch(es) at altitude 35000

   Made 59 corrections to 25 ft jumps in cruise
```

Smoothed radar and ADS-B record comparisons for position and altitude are shown in Figures 6 and 7: the planview (x-y) in Figure 6, the altitude in Figure 7. While there were no jumps in the cruise portions of the Mode-C raw altitude shown in Figure 3, jumps did occur in the ADS-B Mode-S record. Appendix B describes the routine *removeAltJumps* that addresses discretization problems. Jump-removal results are shown for the Mode-S data in Figure 8. Note that *nJ = 59* is the count of 25-ft jump corrections. Note also that the altitude plots shown in Figure 7 seem to deviate below 18000 ft. Perhaps the Mode-S altitude was not corrected for terminal-area pressure conditions.

A comparison of smoothed radar and ADS-B ground speed is shown Figure 9, and of ground track in Figure 10. While the position records match each other quite closely, the velocity records are not quite as close a match. Artifacts in the smoothing of (mostly) nominal 12-sec radar position are probably responsible. However, there is no delay in the smoothed velocity estimates. There was no information about climb rate in the data base, although climb rate is available from the on-board GPS receiver.

It was mentioned earlier that radar records available to TSAFE include real-time filtered estimates of ground speed and ground track. These estimates, especially those derived from ARTCC radar, are generally not accurate and show unacceptable lag for use with a tactical conflict-resolution tool. This can be observed by comparing the filtered radar and ADS-B velocities, as shown in Figures 11 and 12. The lag in the Center portion of filtered ground track shown is seen to be greater than 36 sec (3 samples).

Finally, the results shown in this example are typical of those observed in other flights taken from the sample that had both radar and GPS tracking. It will be interesting to determine if the accuracy and high sample rate of GPS tracking leads to the expected improved ability for TSAFE to detect and resolve aircraft conflicts.

VI Concluding Remarks

This document has described a program for smoothing NAS radar data to simulate tracking data downloaded from an ADS-B equipped aircraft. The purpose was to provide a track-data source for testing TSAFE in an ADS-B environment. A comparison of smoothed and GPS-derived tracking data showing the correspondence between aircraft ground speed and ground track estimates has been presented.

It should be emphasized that the smoothing code described here is quite modular and could be readily adapted for other applications that require smoothing of radar-track data.

Appendix A

Figures 2 through 12, cited in the smoothing example discussed in Sections IV and V

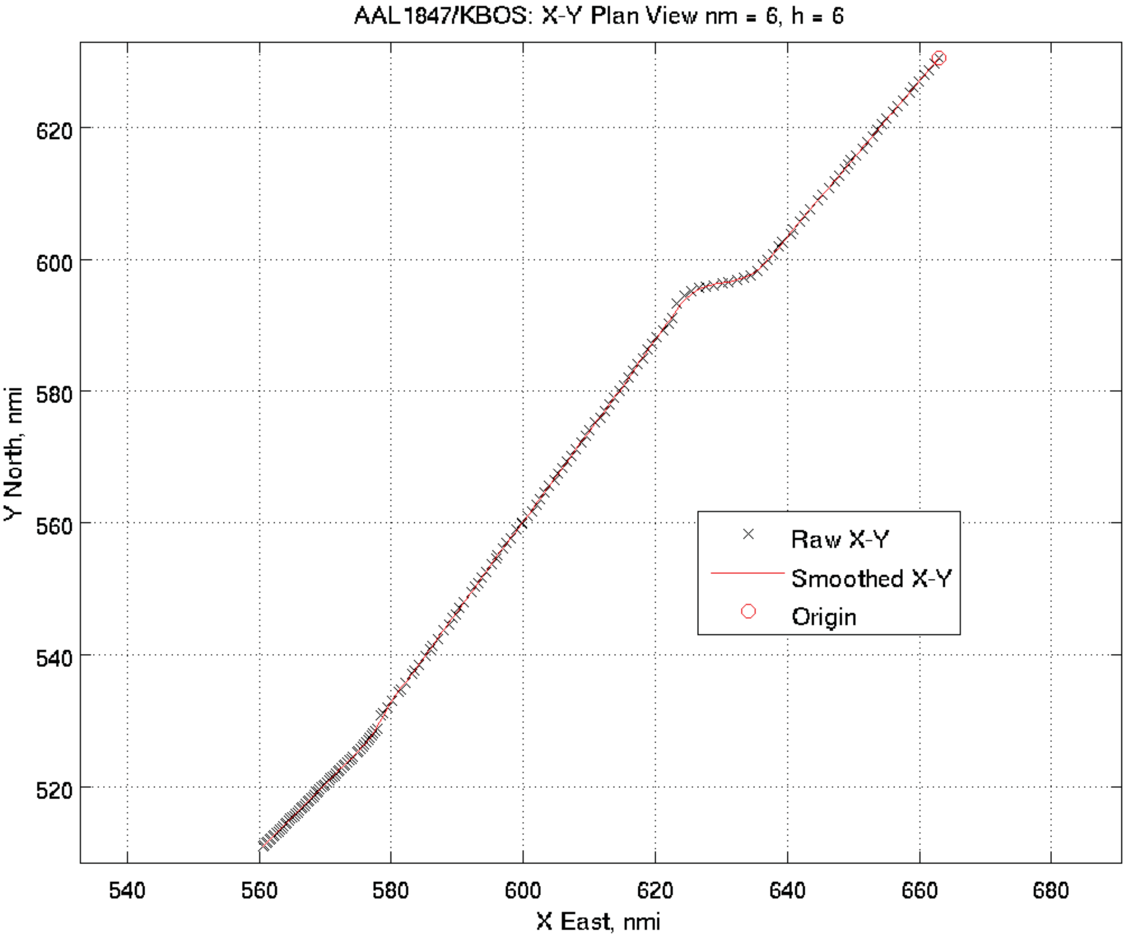

Figure 2. Planview display of raw and smoothed X-Y tracks for AAL1847/KBOS.

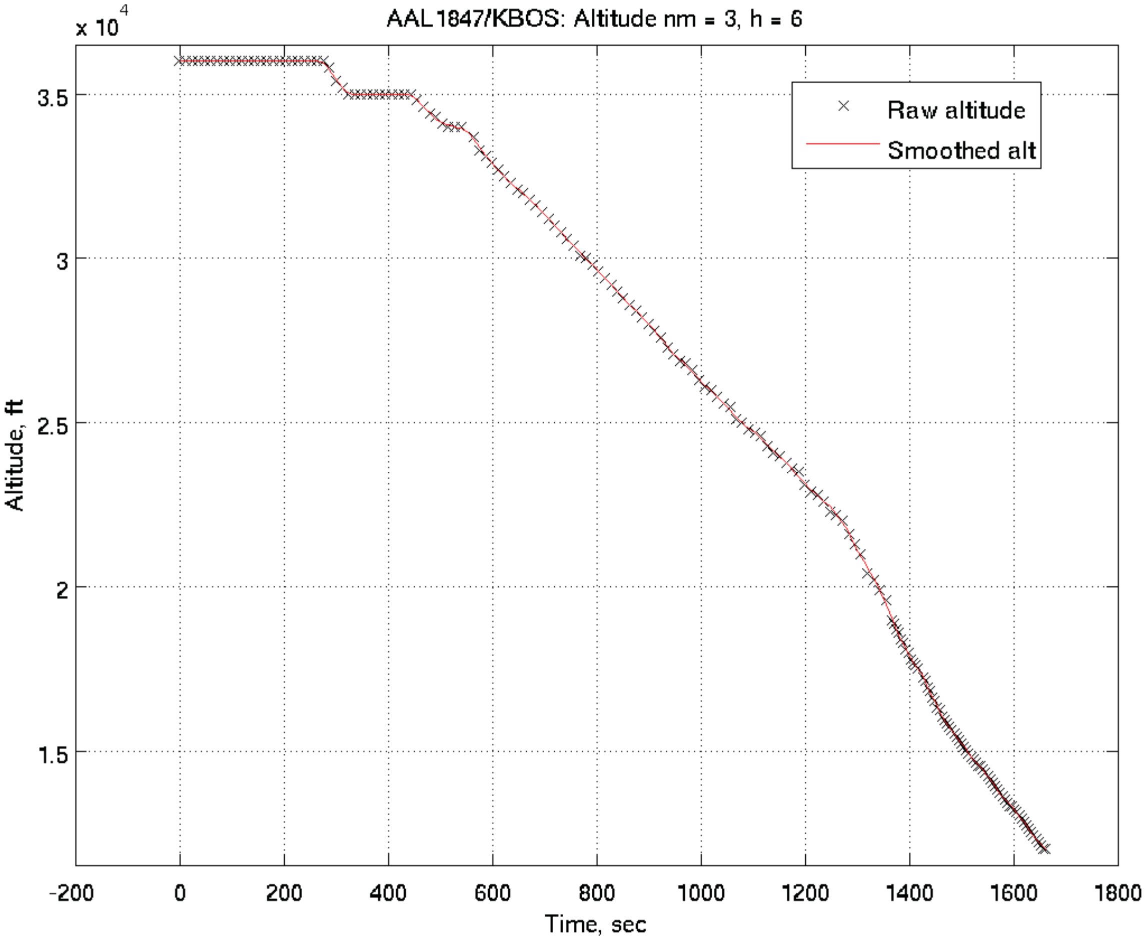

Figure 3. Time histories showing raw and smoothed altitude for AAL1847/KBOS.

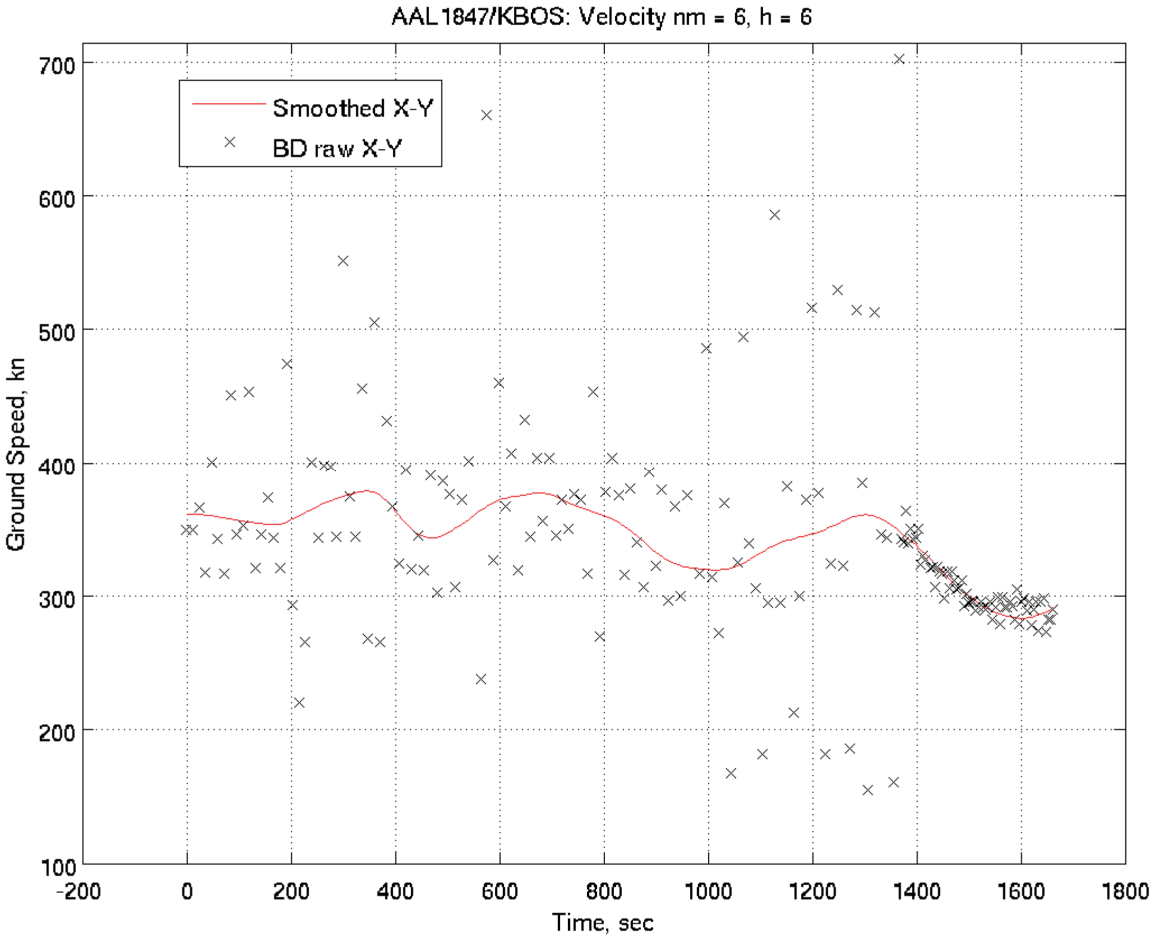

Figure 4. Ground speed histories from back-differenced and smoothed X-Y tracks.

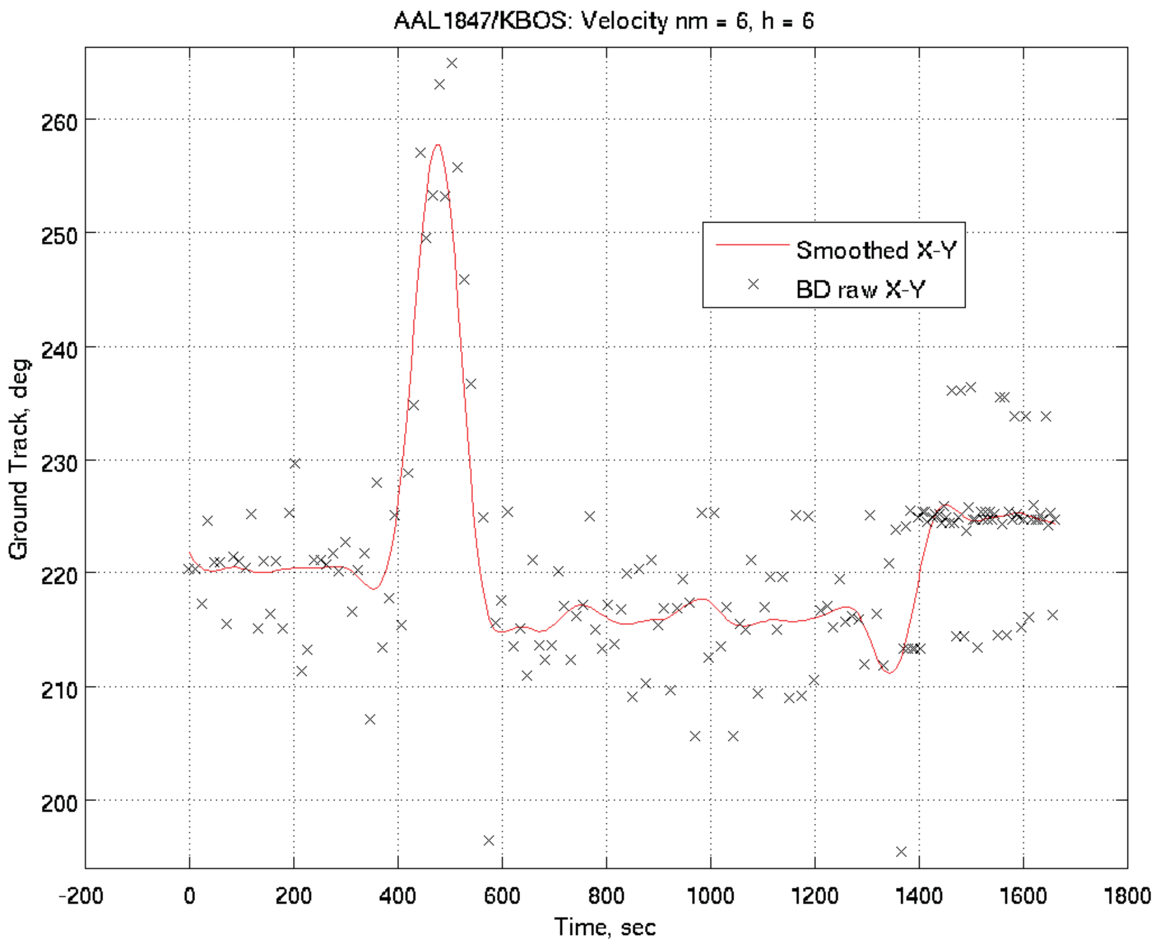

Figure 5. Ground track histories from back-differenced and smoothed X-Y tracks.

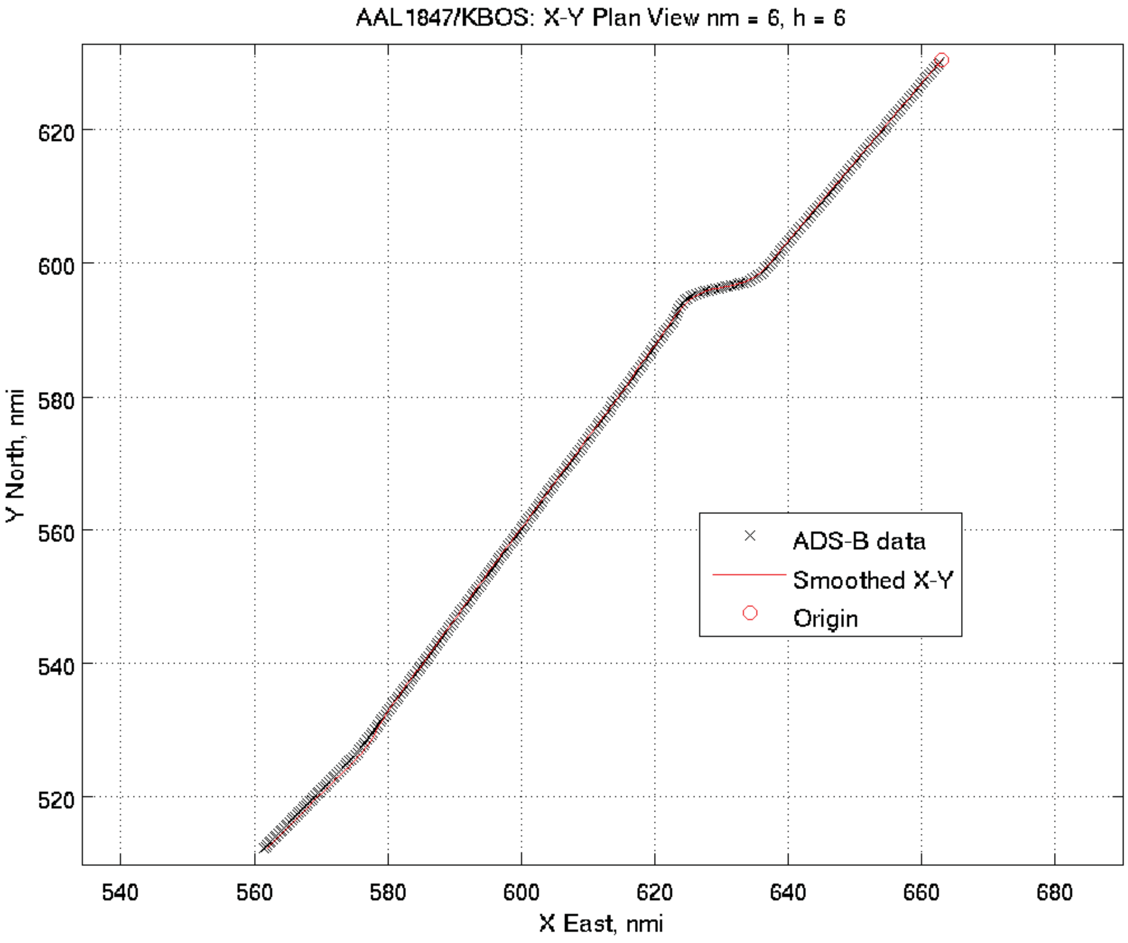

Figure 6. Planview display of ADS-B and smoothed X-Y tracks for AAL1847/KBOS.

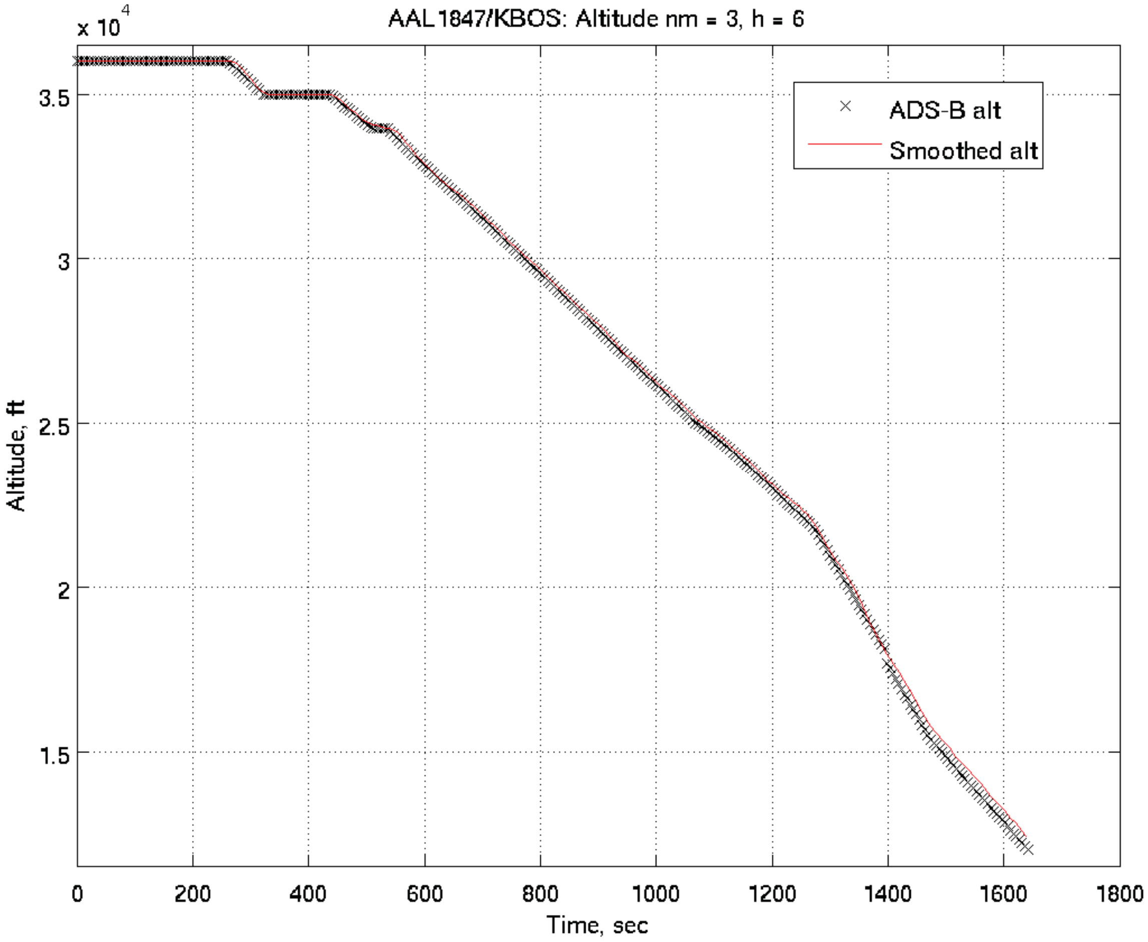

Figure 7. Time histories showing ADS-B and smoothed altitude for AAL1847/KBOS.

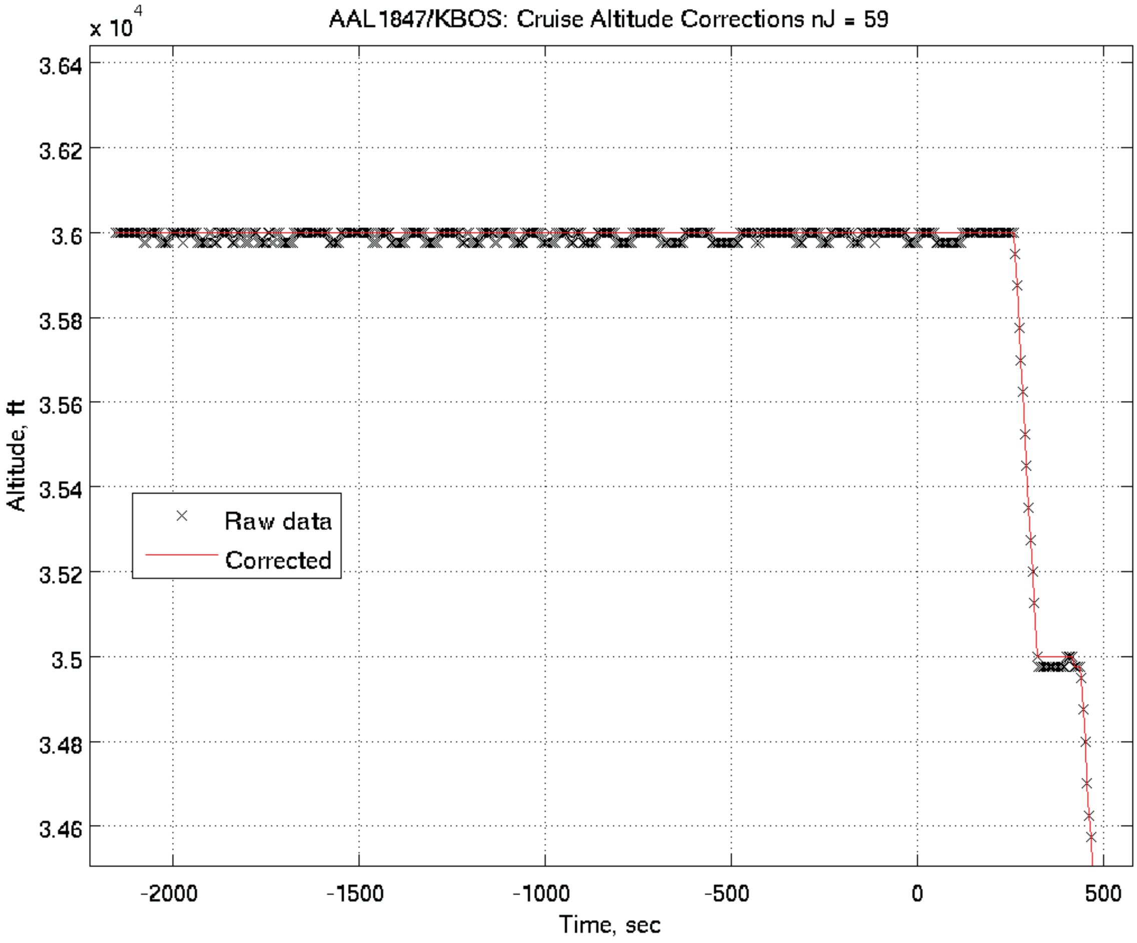

Figure 8. Display of 25-ft jump corrections in cruise for ADS-B Mode-S altitude.

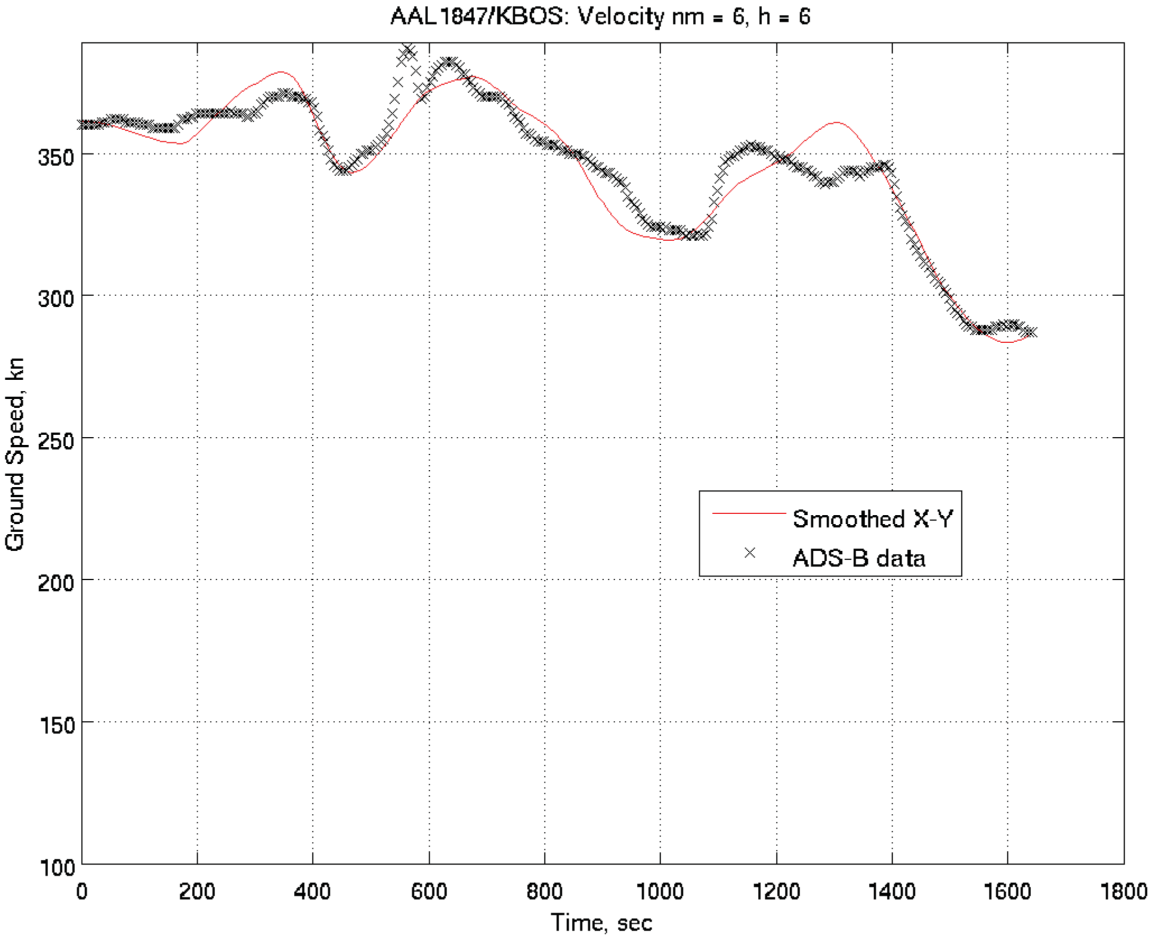

Figure 9. Comparison of ground speed from smoothed radar X-Y tracks and ADS-B data.

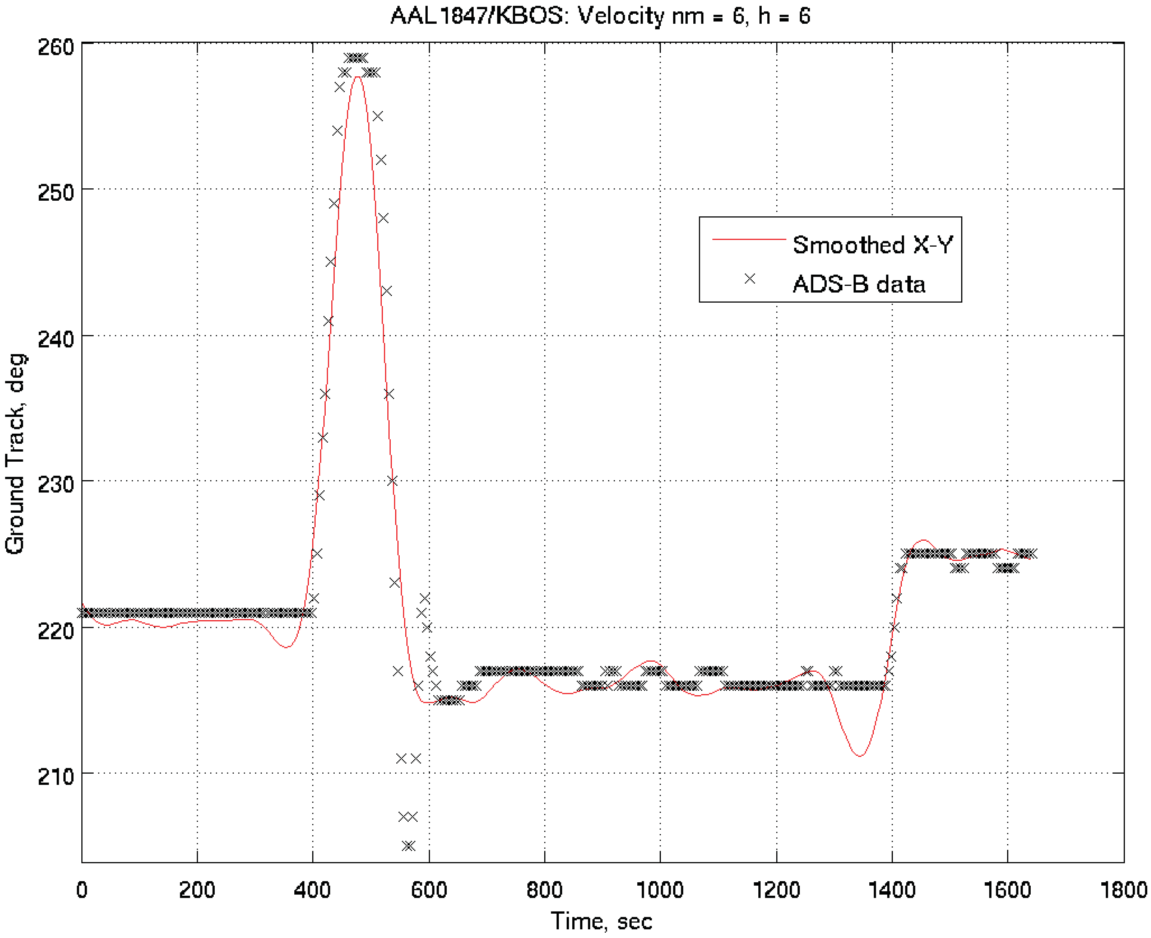

Figure 10. Comparison of ground track from smoothed radar X-Y tracks and ADS-B data.

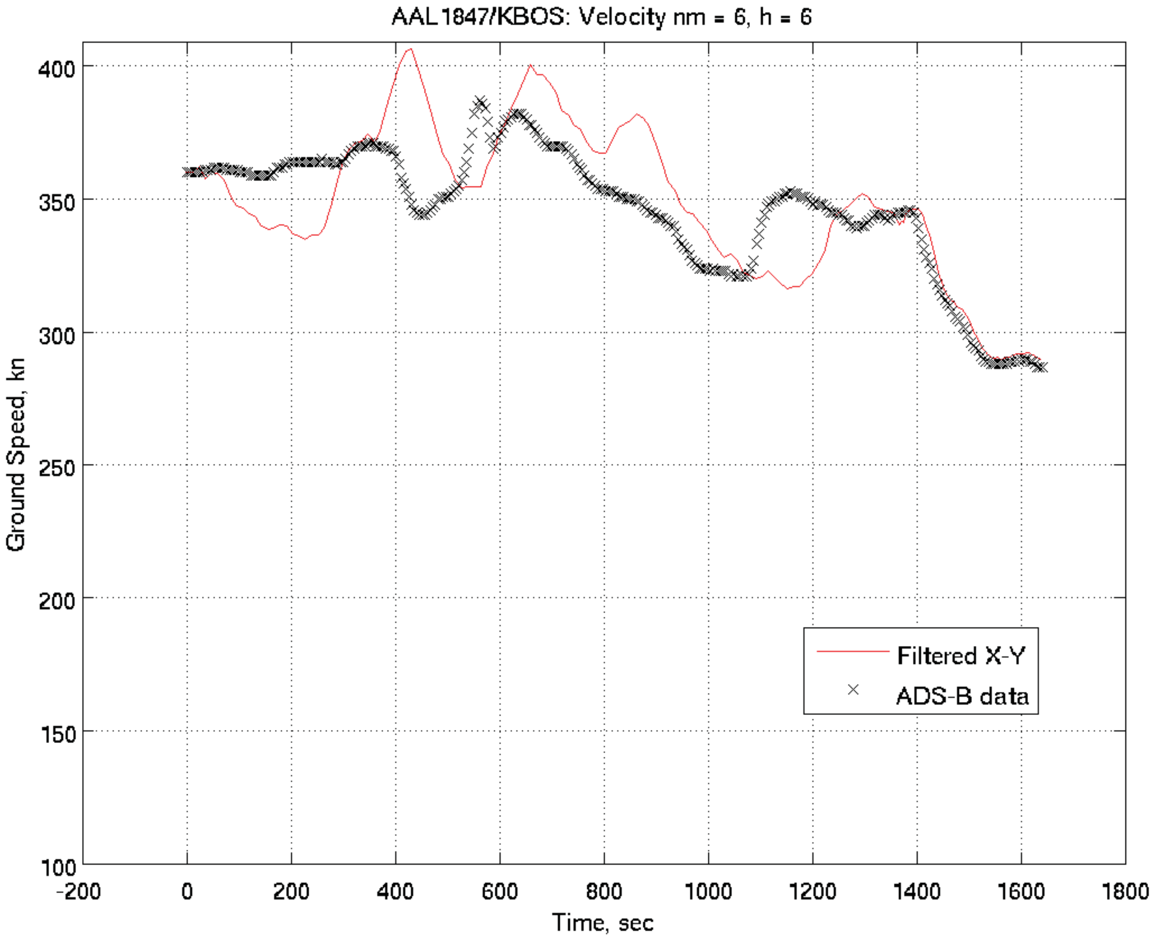

Figure 11. Comparison of ground speed from filtered radar X-Y tracks and ADS-B data.

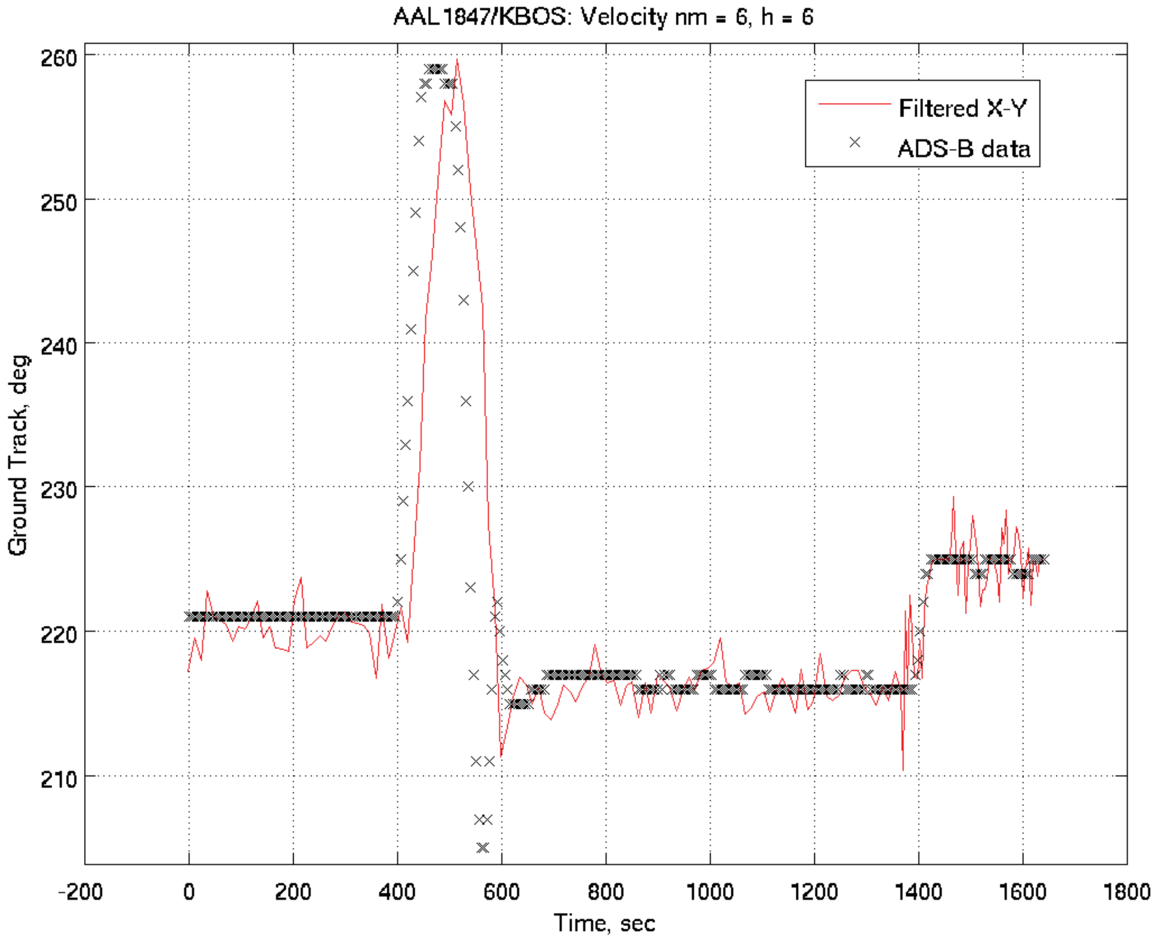

Figure 12. Comparison of ground track from filtered radar X-Y tracks and ADS-B data.

Appendix B

Core Routines

There are two Matlab programs that generate smoothed x, y, and altitude records from TSAFE radar tracks. The first, *batchSmooth*, works only in “batch” mode: it reads track records from file *TSAFE.in.unsmoothed*, smoothes the track data, replaces the original tracks, and stores the new file in file *TSAFE.in.smoothed*. There is very limited screen output and no plots are provided.

The second program, *smoothTracks*, allows the user to specify the aircraft whose tracks are to be smoothed. This program can also be used in batch mode. However, if a single aircraft is selected, screen and printed output (including plots) are provided. This mode is useful for analyzing a particular case and for code debugging.

Both *batchSmooth* and *smoothTracks* utilize the same core routines. These are described below, in the order in which they are called.

getACIDList reads flight registration messages[6] from *TSAFE.in.smoothed*. These correspond to filed flight plans. An example of the format required is given by

FLT 1341581430 AAL1225/KDFW MD82/Q IFR RVSM DEP 320 454

Here fields 2 and 3 are the record timestamp and the call sign of the flight to which it applies. Field 4 gives the FAA aircraft type designation. Field 5 indicates the filing status of the flight as IFR (Instrument Flight Rules). Field 6 indicates that the aircraft is equipped for RVSM (Reduced Vertical Separation Minimum). Field 7 is the ATC type, in this case “departure”. Fields 8 and 9 are the filed altitude in units of 100 feet and the filed airspeed in knots. The routine keys on “FLT” and compiles a list of unique aircraft call signs that will be used to access track data. *getACIDList* is called by *batchSmooth* and *smoothTracks*.

getRawTRKData reads track messages[6] for a given call sign. Each message provides time, x, y, altitude, ground speed, ground track, and sector. An example of the format required is

TRK 1341587269 AAL1225/KDFW -3.2 485.247 476.341 22400 443.2 261.6 1281.5 100

Here field 2 is the record time and field 4 is the track time, relative to the record time. Because a delay of a few seconds can occur between the actual track and the recording of the data, the actual track time can be a few seconds earlier than the record time. The actual track time is obtained by adding the relative track time to the record time. Fields 5 and 6 give the x-y coordinates of the aircraft position in nautical miles (nmi) with reference to the stereographic coordinate system of the Center. Field 7 gives the Mode-C barometric altitude of the aircraft in feet (ft). Fields 8-10 give real-time estimate of ground speed in knots (kn), ground track in degrees (deg), and climb rate in feet per minute (fpm). Field 11 contains the sector number. The routine keys on “TRK”and “AAL1225/KDFW,” creates a track file for altitude greater than *minAlt*, and calculates the average sample time. *getRawTRKData* is called by *batchSmooth* and *smoothTracks*.

trkTransInfo logs transitions between center radar hits (nominal 12-sec samples) and terminal radar hits (nominal 4.8-sec samples). Called by *getRawTRKData*.

removeAltJumps removes jumps in raw altitude during cruise associated with altimeter discretization (usually 100 ft). Their removal enhances altitude smoothing without compromising conflict detection. Called by *getRawTRKData*.

findDiscretLevels determines the altitude discretization level (default 100 ft). Called by *removeAltJumps*.

procTRKData performs wildpointing, interpolation, and smoothing of altitude, x (East), and y (North) tracks for one aircraft raw radar data record with nominal 12-sec and 4.8-sec samples, and calculates altitude rate, ground speed, and ground track. Called by *batchSmooth* and *smoothTracks*.

findStatWP_XY; *findStatWP_H*: these routines apply a statistical test to identify and remove wild points from the raw records. Wild points in x and y tracks are assumed to be correlated, as they are derived from radar returns. The aircraft altitude record is derived from a Mode-C or Mode-S transponder, and, for wildpointing, is considered separately from the x-y records. Called by *procTRKData*.

checkRecForWPs smoothes a single raw record and applies a three standard-deviation error limit to identify wildpoints. Called by *findStatWP_XY* or *findStatWP_H*.

smoothPass does "moving-arc" (cubic) smoothing. The *nm+1* points at the start and end of a record are determined by fitting a cubic to the first and last *narc = 2*nm+1* points. Interior points are determined, one by one, as the center of each *narc*-point cubic moved through the record. Called by *checkRecForWPs*.

cubicFit fits a cubic to the narc point array *Zp* at time points in the array *Tp*. The center point of the fit and time derivative are returned, unless *Tp* contains the first or last *narc* points. In those two cases, the first and last *nm+1* points of the fit, respectively, are returned. Called by *smoothPass*.

removeWPs removes the wild points in records x, y, and altitude found by *checkRecForWPs*. Called by *findStatWP_XY* or *findStatWP_H*.

interpRec does "moving-arc" (cubic) smoothing as in *smoothPass* on the wild-pointed record and provides interpolants of period *hFin* to span the record. Called by *findStatWP_XY* or *findStatWP_H*.

cubicFitInt fits a cubic to the *narc* point array *Zp* at time points in the array *Tp*. The points returned are interpolants of period *hFin* for times less than or equal to the rounded value of the center point of the *Tp* array, until *Tp* contains the last *narc* points. Then the points returned are interpolants for times less than or equal to the rounded value of the last point of the array. Called by *interpRec*.

smoothTRKRec controls the smoothing of each x, y, and altitude record that earlier was wild-pointed and interpolated to *hFin* seconds. There are two passes performed to provide a smoothed estimate and its time derivative. Called by *procTRKData*.

smoothPass: see earlier description for this routine. Called by *smoothTRKRec*.

cubicFit: see earlier description for this routine. Called by *smoothPass*.

GetRateEstimates uses time derivatives xDot, yDot, and hDot to calculate estimates of ground speed (kt), ground track (deg), and climb rate (ft/min). Called by *procTRKData*.

References

[1] Refer to the website: www.mathworks.com/help/matlab/

[2] Refer to the website: www.faa.gov/nextgen/implementation/programs/adsb/

[3] Russell A. Paielli: "Evaluation of Tactical Conflict Resolution Algorithms for Enroute Airspace," AIAA Journal of Aircraft, vol. 48, no. 1, Jan-Feb 2011.

[4] Russell A. Paielli, Heinz Erzberger, Danny Chiu, and Karen Heere: "Tactical Conflict Alerting Aid for Air Traffic Controllers," AIAA Journal of Guidance, Control, and Dynamics, vol. 32, no. 1, Jan-Feb 2009.

[5] Russell A. Paielli: "Tactical Conflict Resolution using Vertical Maneuvers in Enroute Airspace," AIAA Journal of Aircraft, vol. 45, no. 6, Nov-Dec 2008.

[6] Russell A. Paielli, Ralph E. Bach: "TSAFE Interface Control Document," NASA TM-216034, Jan 2013.

[7] Refer to the website: https://atmweb.arc.nasa.gov/dw

[8] Thomas S. Logsdon: "Moving-Arc Data Smoothing for Flight-Analysis Programs," 1st AIAA Annual Meeting, Paper No. 64-324, Jun 1964.

[9] For access to the code, contact Russell.A.Paielli@nasa.gov