RTO-57
Traffic Flow Automation System (TFAS) Analysis Report

September, 2001

Jesse Clayton
James Murphy

Abstract

Decision support tools are used by Center air traffic controllers to help monitor and mitigate the flow of traffic over the continental United States (CONUS). The Enhanced Traffic Management System (ETMS) provides controllers and traffic management units (TMU's) with tracking information, as well as predictions of aircraft trajectories and airspace demand. This predictive data promotes long term planning and strategy. However, certain features of the current implementation have been shown to be qualitatively insufficient to be used as reliable aids in the traffic management process.

The Traffic Flow Automation System (TFAS) is being developed to improve the accuracy of trajectory and Monitor Alert predictions made by ETMS. Monitor Alert is an indication of predicted airspace demand exceeding capacity. Preliminary studies have suggested that the TFAS trajectory synthesizer using TFAS airspace adaptations produces better trajectory predictions than the ETMS trajectory engine, and therefore better predictions of airspace demand and Monitor Alerts. Improvement of Monitor Alert predictions may help controllers plan for future events, maintain a balance between airspace capacity and demand, and reduce air and ground delays throughout the system.

This document represents an analysis of the predictive capabilities of TFAS, and how those predictions compare to ones made by ETMS under similar conditions. Included are studies of the trajectories themselves, and of information derived from those trajectories (ie sector counts). Also presented are projections of TFAS performance under operational conditions. Though more studies are required, current results show that TFAS tends to make better trajectory and sector count predictions for events occurring within 35 minutes of the prediction, while ETMS tends to perform better in longer look-ahead situations. Initial results also show that using the Host instead of ETMS as the primary data source for TFAS results in better predictions, such that TFAS performs better than ETMS for up to 50 minute look-ahead times.

This document contains a detailed description of the methods and materials used to perform the analyses. The appendices include among other things complete listings of all necessary raw data sources to replicate the study. The results of the analyses are also presented, and should serve as a baseline for future studies of TFAS.

1 Introduction

The purpose of these analyses is to do a direct comparison between TFAS and existing decision support tools. As ETMS is currently used operationally, most of the data presented in this document compare TFAS directly to ETMS. In some cases, we present several sets of TFAS data together, showing either TFAS running under different conditions, or projections of how TFAS would run with operational or algorithmic improvements.

Like ETMS, TFAS was designed as a national system. However, TFAS is composed of 20 instances of the Center TRACON Automation System (CTAS), one for each CONUS Air Route Traffic Control Center (ARTCC). To simplify the analysis process we elected to use only one of these subsystems in our comparison. As this document will serve not only as an analysis report, but also as a baseline for future studies, in several places we note how the analysis might be conducted differently in "national" mode.

2 Analysis Methods and Materials

In this section we describe the raw data sources and the processes used to conduct the analyses in this study. Presented first is a summary of the necessary steps for obtaining and preparing the required data. Here we will describe the general format and sources for such raw data, as well as the programs written to process it. For these programs, a functional overview is provided in the body of the document, and the actual invocation and use is covered in the appendices.

The following describes the characteristics of the actual datasets used in the analyses. Resources for raw data and descriptions of the actual data sets is appears at the end of the section.

We break the analyses performed into three categories:

- Trajectory Prediction Accuracy
- Sector Load Prediction Accuracy
- Projections of TFAS Predictions under Operational Circumstances

Trajectory prediction accuracy refers to the accuracy with which a tool is able to predict the positions of specific aircraft at some point in the future. This category includes sector entry time prediction comparisons and sector duration prediction comparisons. Sector load prediction refers to ability of a tool to predict accurate counts of aircraft within some region of airspace. The region is generally an Air Route Traffic Control Center sector, and the measurement is known as "sector demand prediction". In this category we include sector count prediction comparisons and sector hit rate.

The third category refers to projections of TFAS predictions under operational circumstances; for example, factoring into the analysis the benefits of the arrival scheduling capabilities of TFAS, and observing the resulting trajectory or sector load predictions.

While each of these categories represents different studies, the data sources and the methods for conducting each are similar. We describe in detail the analyses in the order of the categories listed above, along with methods and materials. These will be preceded, however, by a description covering the procedure to obtain the proper input data sets for the analysis tasks, as well as the processes used to collect the raw data. We will cover any necessary deviations from this setup procedure (for special analyses) at the appropriate point in the discussion.

We conducted data all of the analyses in this study using two sets of raw. After the general setup and pre-processing tasks have been discussed we will present the general characteristics of each data set.

2.1 General Setup and Analysis Pre-processing

In this study, we compare the capabilities of TFAS and ETMS to make accurate predictions for trajectories and sector loading. In fact, all of the analyses in the study deal directly with or are a derivative of trajectory predictions.

For an aircraft, trajectory prediction is the ability to calculate the position of that aircraft at some time in the future, or to predict "events" for that aircraft. An event could define any position at any time; however, since a primary focus of this document concerns ARTCC sector demand, we look at the events of aircraft entering and exiting sectors. For this reason, we developed a pre-processing procedure that organized the data sets to make the comparison of sector entry and exit predictions a relatively simple task.

For any such analysis in this domain, we define three required datasets. These primary datasets will be the input to of all of the analyses conducted for this document:

- TFAS Sector Entry and Exit Predictions
- ETMS Sector Entry and Exit Predicitons
- Actual ("truth") Sector Entries and Exits

The next section concerns the characteristics of the raw data sources and the procedure to obtain the above information from those sources. In some cases, data processing and filtering software has been developed by the analysts to aid in these tasks. In such cases, the algorithms are described in the text, and the actual usage of the software listed in the appendices.

2.1.1 Obtaining Prediction Data

Since the results of this study focus on a comparison of TFAS and ETMS, it is important that TFAS and ETMS share the same data source when conducting this analysis. ETMS receives host data from each ARTCC on a 60 second update rate. It also receives TRACON and oceanic data at various rates. ETMS uses this data to calculate the trajectories it uses to predict events, but also forwards position and route data to each field site. Through a process known as ftm_connect, the TFAS system also receives these data in realtime. Ftm_connect is used as the primary data source for TFAS running in "live mode".

TFAS Prediction Data

For analysis, however, we found it important to have archived a data set that could be replayed multiple times for conducting successive studies. For TFAS, this data set comes in the form of "orig" files. In addition to sending information through ftm_connect, ETMS records the same information in files known as "orig" files. The data in these files is in the same format as that in ftm_connect, so it was a logical choice to use as an archived TFAS data source. In fact, before the development of ftm_connect, TFAS was built to use orig files as its primary data source. The orig files contain the track and flight plan information necessary for TFAS to make trajectory predictions. The process of obtaining prediction data from orig files can be broken down into several steps. Throughout this procedure, several intermediate data sets are maintained and used as input to future analysis steps.

Creating the Playback File

TFAS inherited from the CTAS the ability to record and playback data from any source for which the processes are configured to accept. Orig files represent just such a source, and the process of transferring this data to “cm_sim” (also known as “playback”) files is relatively simple. It is necessary to create this second tier of recorded data (the cm_sim files) for reasons described later.

Three components of TFAS are needed to generate this file. These are the ETMS Data Acquisition Routing (EDR), the Input Source Manager (ISM) and the Communications Manager (CM). EDR is responsible for reading and processing orig files, ISM handles the communications between EDR and the CM, and CM writes out the cm_sim file. These three processes can be started in any order. EDR is invoked with the following options:

```
edr –dir <origdir> -orig <filename>
```

where

`<origdir>` refers to the path to the directory where orig files are stored

`<filename>` refers to the orig file with which to begin processing

ISM is invoked with the following options:

```
ism –data <adapt dir>
```

where

`<adapt dir>` refers to the name of the adaptation to use for this analysis

The TFAS adaptation directory should appear in the relative path “../../adaptation/<adaptdir>”. CM is invoked with the following options:

```
cm –data <adapt dir> -add_all_fps
```

where

`<adapt dir>` refers to the name of the adaptation to use for this analysis

The “-add_all_fps” indicates that cm should use all tracks and all flight plans for this analysis (as opposed to using arrivals only).

After the three processes have started, the CM graphical interface is used to connect CM to ISM and then to EDR. Upon completion of this task, CM will begin to receive data processed from the orig files. One orig file generally contains one hour’s worth of data. In order to perform runs longer than an hour, the respective orig files are moved into the same directory (the directory specified in the –dir option to EDR). Note that ETMS names orig files with the convention “orig.mmddhhmmss”; EDR uses the portion of the file name appearing after the ‘.’ to determine replay time and correct order in which to open files. Attempting to open an orig file with a different or corrupted naming convention results in undefined behavior. EDR is configured to open each properly named orig file in the correct order, closing each as it is finished and looking for the next. When EDR reaches the end of the last orig file, it will remain running but discontinue processing (as there is no data to process).

It should be noted that orig files do not retain a continuous record of flight plans throughout the data set. The flight plan for a specific flight will appear somewhere in the orig data within the 24 hours before the filed time of departure. TFAS, however will not process track data (and will not make trajectory predictions) for flights without a known flight plan. In order to build up the TFAS flight plan database, EDR, ISM and CM should be started with data starting 12-24 hours before the time period over which the analysis will occur. For example, if an analysis is to be performed starting at 0700 on June 10 and ending at 0700 on June 11, orig file reading should begin with data from at least as early as June 9 at 1900.

After connection to EDR has been established the graphical interface of the CM is again used to begin recording of the cm_sim file. Unlike the orig data, it is not necessary to begin recording earlier than the desired analysis period.

When EDR has finished reading all available orig data, the CM is used to stop the recording and disconnect the other processes. The resulting cm_sim file will have been populated with track and flight plan information, however the orig files do not contain any wind prediction data. This information must be processed, and the cm_sim file edited to add calls for TFAS to read wind data when the analysis is replayed.

Processing RUC Data

The wind data used by the Trajectory Synthesizer (TS) process in TFAS comes from the Rapid Update Cycle (RUC) data provided by the National Weather Service (NWS) National Center for Environmental Prediction (NCEP). For an operational CTAS system, the Weather Data Processing Daemon (WDPD) is notified when the latest RUC data has been downloaded from NCEP. The RUC data is provided in 1, 2, 3, 6, 9, and 12 hour predictions, and it also contains a file with the current wind information. However, due to the length of time it takes to generate the files and transfer the data, the 2 hour forecast is generally used for CTAS processing. When WDPD has been notified that new RUC data is available, it reads in the data, processes the information into the binary format used by TS to calculate the extrapolated wind data for any 3 dimensional position in the airspace defined by the adaptation.

For example, on 29 August 2001 at 1320 GMT, the WDPD is notified that the RUC data based on the 1200 GMT winds is available. WDPD reads in the file called ruc2.T12Z.grb2f02. The name of the file has meaning. The T12Z specifies that it comes from wind data valid for 1200 GMT. The grb2f02 specifies that it is 40km wind data forecasted for 2 hours, i.e. 1400 GMT. WDPD parses the wind file, reads the time and date information from internal data, processes the information into the CTAS binary format and creates a file called 2001_241_12_00_02.bin. As can be seen, the name of the file contains all f the information needed by CTAS to determine when the wind information in this file is valid. The CM is notified by WDPD when the file is ready and the new wind data is sent to each of the TFAS processes that need it.

For our analyses, we do not run with a live system. The RUC data was archived for the specific dates of our analyses by the North Texas (NTX) CTAS facility and provided to us for processing. Each CTAS subsystem of TFAS works the same way a stand alone version of CTAS. Since we are working with archived data, WDPD was used to pre-process each of the RUC files needed for analyses. In order to emulate the predictions of a live system, the 2 hour forecast was used for each wind update.

WDPD is already capable of processing specified ruc files to produce TFAS binary files. However, this is only for testing purposes, and it does not name the files with the appropriate naming convention. Modifications to WDPD were made for the purposes of this analysis to force WDPD to name the binary files with the appropriate naming convention while running in test mode. Depending on the analysis being preformed, the wind data covering each ARTCC airspace used in the analysis will need to be created separately. The binary files only cover the airspace defined by adaptation for each subsystem of TFAS.

WDPD is invoked as follows:

```
wdpd –data –ff <ruc file> -grid_res <m> –grid <n> –out_dir <dir>
```

Where

`<ruc file>` is the name of the ruc file to read in
`<m>` indicates the ratio of output to input grid points
`<n>` indicates the type of input RUC file
`<dir>` is the name of the directory to write the output bin file

For our analysis, a grid resolution of "2" (-grid_res option) and a RUC file type of "1" (-grid option) were used.

The archived bin files are stored in directories for reading into the CM at the appropriate time. If running a single center TFAS analysis, the input of aircraft into the system is done through archived cm_sim files. The calls for CM to read new weather are embedded in the cm_sim file at the appropriate time. If the cm_sim file is created from archived ETMS data, the calls may need to be inserted by hand.

In order to allow for multiple ARTCC TFAS analyses, modifications were made in CM to read in the preprocessed binary files at the appropriate time indicated by the binary file name. This convention will also work in the single ARTCC analysis, if needed.

Adding Weather Calls to the Playback File

In order to obtain accurate trajectory predictions, the created cm_sim file will be replayed through TFAS, with the additional wind information found in the bin files. The cm_sim file must be edited to add the appropriate calls to read weather. This can be done by adding a line of the following format at the appropriate places in the file:

```
WTHR_BIN_FILENAME <rel time> <type> <filename> -NS-
```

where

`WTHR_BIN_FILENAME` is the keyword that tells CM to read a weather file
`<rel time>` is the relative time from the start time of the playback to read the weather file
`<type>` refers to the type of bin file to read
`<filename>` refers to the name of the bin file with full or relative path
`-NS-` sets an option not used in this analysis to "NOT_SET"

An example of a bin file filename (without path) is

```
2001_240_21_00_02.bin
```

This name is interpreted as a file of wind data predicted at the top of the 21st hour of the 240th day (Julian calendar) of the 2001st year. The "02" near the end of the filename indicates that this was a two hour prediction. The call to read this file, then, should appear at or around the line in the cm_sim file corresponding to the 23rd hour of the 240th day of the 2001st year (23rd hour because 21st hour plus 2 hour prediction). A similar line should be entered at or near every hour throughout the cm_sim file[1]. However, as the file is long and the information in it unwieldy, we developed a simple 'C' program, "Insert Weather", to aid in this task. The details of this program are covered in the appendices.

---

[1] This describes the process used to insert weather calls in the "new" format of the cm_sim file. Later in the section, a process is described to enter the appropriate calls in cm_sim files obtained from operational CTAS. These operational files, incidentally, are in the "old" format.

The resulting cm_sim file (the original with weather calls added) should be used as the input file for the next step of the analysis, in which event predictions are actually produced.

Obtaining Sector Entry and Exit Time Predictions

TFAS is run again to obtain sector entry time predictions. Three of the TFAS processes are needed to produce this information: the CM, the Route Analyzer/Trajectory Synthesizer (RA/TS) and the TFAS Database Process (TDP). The CM is responsible for reading and processing the recorded cm_sim file (and weather data) and producing event lists (event predictions), the RA handles trajectory prediction requests from the CM, and the TDP, though not serving any computational purpose, is required to obtain the event lists. CM is invoked with the following options:

```
cm –data <adapt dir> -add_all_fps –tfas <tdp machine>
```

where

`<tdp machine>` refers to the name of the machine on which the tdp will run

The other options were described in the previous section.

RA is invoked with the following options:

```
ra –data <adapt dir> -ts <ts exe>
```

where

`<adapt dir>` refers to the name of the adaptation directory

`<ts exe>` refers to the name (with path) of the TS executable

During startup, RA creates a fork and starts the TS as a separate process,. These communicate with each other via shared memory.

TDP is invoked with no options:

```
tdp
```

TDP should be started first, followed by CM and RA. CM will attempt to connect to TDP at startup, and RA will attempt to connect with CM. Processing of the cm_sim file is started through the graphical interface of the CM. Output of the event list predictions is started through the dynamic debugging window (accessed by pressing “Shift F3” in the CM). The debug command to send is as follows:

```
CM CMD_SECTOR_CROSSINGS <file name>
```

where

<file name> refers to the file to which to write the event list predictions

Writing of event lists can be stopped by sending debug command

```
CM CMD_SECTOR_CROSSINGS -
```

This step results in the production of a data set of TFAS sector entry and exit predictions, stored in an ASCII file. This is one of the primary inputs to the analysis task; the file will be henceforth be referred to as the “TFAS event lists”.

Alternative Data Source: Using Host Data as the TFAS Data Source

We have observed that the TFAS performance increases when Host data is used as the data source for TFAS, and in many analyses we have included the results using Host data. The process to obtain TFAS prediction data from Host data is almost identical to the process described in the previous section.

For this step, TFAS prediction data is obtained using a operational CTAS cm_sim file recorded at the North Texas (NTX) CTAS facility. The first step in this case is to add weather calls to the file. NTX uses a version of CTAS that records the “old” version of the cm_sim file, which is not yet handled by the “Insert Weather” program. However, as these files were recorded in “live mode”, they already contain calls to read bin files. We simply used the unix tool “sed” to change the paths to the bin files from what was used at NTX to a local naming convention .

In order to handle the “old” version of the cm_sim files, the CM must be recompiled with the “-DOLD_PLAYBACK” define passed to the compiler (RA and TDP do not need to be recompiled). With the resulting executable, obtaining prediction data is done exactly as described in the previous section. The next primary input is a data set of ETMS event lists.

ETMS Prediction Data

The ETMS prediction data must be obtained through Volpe. A special process was developed there to dump the event lists for specific flights from the ETMS Flight Database (FDB) into an ASCII file. These files are only available upon request, and will henceforth be referred to as the “ETMS event lists”. For our analysis, the ETMS event lists are obtained as a set of files, each file representing sector (as well as meter fix and airport) entry and exit times (exits in the case of sectors only) for every specified flight in the ETMS flight database (FDB) at the time the file was written. The files are generally created in five minute intervals, providing a robust data set for comparison. The ETMS event lists require no preprocessing and are used as direct input to the analysis. The event lists produced by both tools are to be compared against truth data, that is, the actual sector event lists.

2.1.2 Actual (Truth) Data

For the analyses, we used Host data archived by CTAS. The data is recorded in the CTAS cm_sim file format, which for our purposes is just track and flight plan data and is processed to produce a file of actual event lists

We wrote a program for this purpose, known as the Cm Sim Event Processor (CSEP). The details of this program can be found the appendices. The cm_sim file contains track information for aircraft within radar range of the ARTCC in question. A track hit for a specific aircraft refers to the state information of that aircraft at a particular time. An aircraft has a set of track hits throughout the file that trace its course through the airspace. From this information, sector entry and exit times are produced according to the following algorithm:

1. For each aircraft appearing in the cm_sim file, step through the track hits for that aircraft and determine the first and last track hits occurring in each sector crossed by the aircraft.
2. If the “extrapolate” option is passed to the program, use state information to extrapolate track hits to actual boundary crossings, and use these for sector entry and exit times. Otherwise, just use the first and last track hits for sector entry and exit times. Call this list of sector and entry and exit times the event list.
3. If the “filter_update” option is passed to the program, eliminate from the event lists any entry times where the previous track hit is nonexistent or difference between the previous track hit and the current track hit is greater than a user specified update rate.
4. Read a file “reject_ac” containing types of aircraft to reject from the analysis.
5. Write the aircraft id’s and event lists to a file “actual_events”.
6. If an aircraft type matches a type in reject_ac, do not write it or its event lists to “actual_events”. Write its id to a file “reject_acids”
7. If an aircraft has an empty event list, do not write it to “actual_events”. Write its id to “reject_acids”.
8. If the “same_arr_dep” option is passed to the program, do not write to “actual_events” aircraft which have the same arrival and departure airport. Write the ids to a “reject_acids” file.

Notes: The "extrapolate" (step 2) option uses a two dimensional ray intercept that tends to introduce significant error to event lists for flights in transition, and is generally not used with host data, since 12 second updates is considered to be accurate enough for the analysis. The "filter_update" (step 3) option is used to filter out aircraft with sparse track hits, or to filter out the first event in an event list. To illustrate, consider the following:

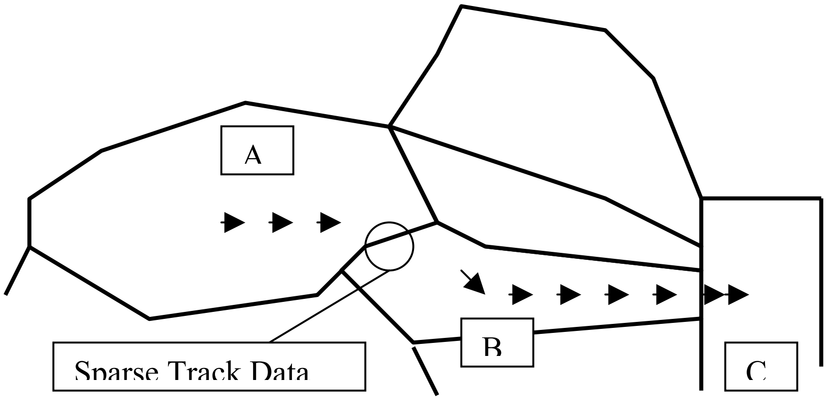


Figure 1 Sparse Track Data in Sectors A and B

The vectors indicate track hits for an arbitrary aircraft. While we can reasonably determine a sector entry time into sector C, the data is sparse enough around the entry into sector B to introduce significant error into that entry time prediction. If the filter update option is set, the entry time into event B would be filtered from the output.

The update rate should be set according to the radar update rate as specified by the data source (in seconds). The program will multiply the specified rate by 1.5, or add 30, whichever results in the smaller value.

The result of this process is a file containing actual sector entry and exit times for aircraft in the data sample. This, along with the two data sets of prediction times, will be used as input data to all of the analyses in this study.

2.2 Actual Raw Data Characteristics

In the previous section we talked about the general characteristics of the raw data used in this study and the steps used to preprocess that data. Here we discuss the actual raw data sets and the sources thereof. These sources are summarized in the appendices. Ideally we would have liked to have several full days of raw data from various centers throughout the US. However, only ZFW host data was readily available, so we elected to focus on that airspace.

All orig data files, as well as ETMS event lists, ware collected through request from Volpe. Every orig file represents one hour's worth of data, in the same format and with the same timing as would appear in ftm_connect. Every event list represents the state of the ETMS FDB at the time the file was written and usually comes in 5 minute increments. We will discuss the deficiencies of such a long time between event lists in each respective analysis as it applies. As the process to dump ETMS event lists is not run as part of operational ETMS, we were required to provide notice ahead of time that we would need the data.

RUC data and operational CTAS cm_sim files are archived by the North Texas (NTX) facility and were provided upon request. RUC data files represent predictions made for some time in the future, usually 2 hours and they generally come in 1 hour increments. A cm_sim file can represent any sample time; all of these files in our analysis represented about 1 day's worth. A summary of these (and other possible) sources of data appears in the appendices.

Raw data was collected for the Fort Worth center (ZFW) for two full days during August, 2001. While our goal was to obtain at least two days of data during relatively light weather, we did not work out all of our problems in obtaining ETMS prediction data (event lists) until early August, 2001, and the two days for which we collected data thereafter experienced significant convective weather. We tried, then, to focus our analysis during periods of the day when the weather was lightest.

August 11, 2001
Orig files: First file starts at 1000Z on August 10, 2001. Last file ends at 1100Z on August 12, 2001. Files for every 1 hour block in between.
ETMS Event Lists: First event list produced at 1100Z on August 11, 2001. Last event list produced at 1005Z on August 12. Event lists for every 5 minute increment in between.
RUC data: First file represents predictions made for 0100Z on August 11, 2001. Last file represents predictions made for 0000Z on August 13, 2001. Files for every 1 hour in between. All files represent 2 hour predictions.
Operational CTAS cm_sim file: First file starts at 1105Z on August 11, 2001 and ends near 1100Z August 12, 2001. Second file starts at 1110Z August 12, 2001and ends near 1100Z August 13, 2001.

The intersection of this raw data occurs from 1105Z on August 11, 2001 to 1005Z on August 12, or 23 hours of data. While the goal was to obtain data for entire days, traffic in ZFW from 1005Z to 1100Z is minimal and should not affect the analysis.

The weather during this sample was relatively clear until about 1830Z, Aug 11. Thunderstorms then developed on the northern portions of the TRACON. At 2045Z a storm cell developed over the west side of Dallas/Fort Worth (DFW), and another at 2130Z. Cells also developed over Dallas Airport (DAL) at the same time. Storm activity diminished around 2300Z. This information came from the DFW ITWS DailyOps report for August 11-12, 2001. The entire reports are included in the appendices.

August 29, 2001
Orig files: First file starts at 0000Z on August 29, 2001. Last file ends at 2300Z on August 30, 2001. Files for every 1 hour block in between.
ETMS Event Lists: First event list produced at 0855 on August 29, 2001. Last event list produced at 0910Z on August 30. Event lists for every 5 minute increment in between.
RUC data: First file represents predictions made for 0100Z on August 29, 2001. Last file represents predictions made for 2200Z on August 30. Files for every 1 hour in between with the exception of the file for predictions at 0400Z on August 29, 2001. This file was missing, but ended up not being part of the intersection of the raw datasets and so would not have been used anyway. All files represent 2 hour predictions.

Operational CTAS cm_sim file: First file starts at 1105Z on August 28, 2001 and ends near 0000Z on August, 29, 2001. Second file starts at 1110Z on August 29, 2001 and ends near 1100Z on August 30, 2001.

The intersection of this raw data occurs from 1110Z on August 29, 2001 to 0910Z on August 30, 2001. Note that earlier in the discussion we stated that it is important to have orig data for at least 12-24 hours before the analysis is to begin so that the CM component of TFAS can add all flights to its flight database. While in this case the difference is only about 11 hours, tests indicated that for this sample an insignificantly small number of flights were affected.

The weather during this sample was relatively clear with the exception of the period between about 2050Z and 2300Z when storm cells developed southeast of DFW. This information came from the DFW ITWS DailyOps report for August 29-30, 2001. The entire reports are included in the appendices.

The next several sections concern the specific information of each analysis performed, as well as the results.

3 Specific Analysis Methods and Materials

This section describes the methods and materials for each analysis in the study and includes results and conclusions. Unless otherwise specified, the three primary data sources for each analysis are those whose collection was described in the previous section; TFAS event lists, ETMS event lists and actual (truth) event lists.

3.1 Trajectory Prediction Accuracy Analysis

The fundamental analysis of which all other analyses in this document are derivatives is that of trajectory prediction accuracy. This capability is the basis of many air traffic decision support tools, and while it is used in each with varying degrees, it is vital to the task of TFAS to predict sector loading and sector monitor alerts.

For an aircraft, trajectory prediction is the ability to predict the position of that aircraft at some time in the future, or to predict "events" for that aircraft. An event could define any position at any time; however, since a primary focus of this document concerns ARTCC sector demand, we look at the events of aircraft entering and exiting sectors. The first analysis of trajectory prediction concerns the accuracy of TFAS and ETMS in predicting sector entry events. While a similar study could be carried out for sector exit events, it stands to reason (and test cases have shown) that this yields no additional insight. The studies are in fact the same, as one sector exit event also represents the entry event for the next sector in the trajectory.

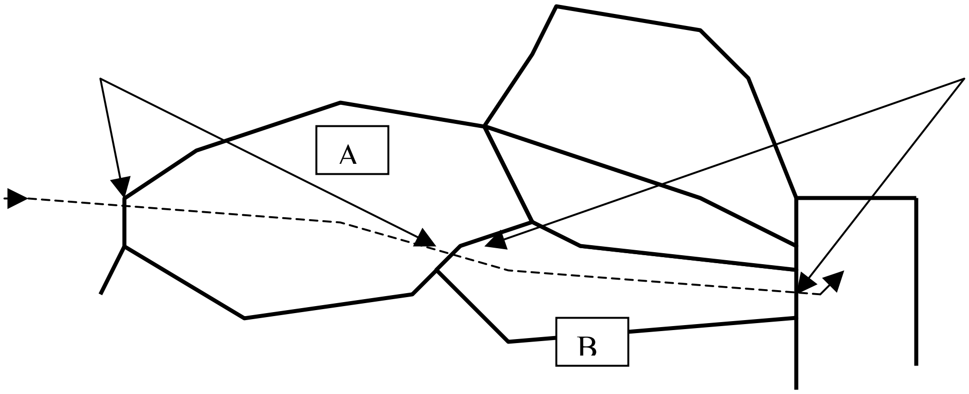


Figure 2 Sector Entry and Exit Events in Sectors A, B and C

In the second analysis we deal rather with the accuracy of TFAS and ETMS in predicting ARTCC sector occupancy time. We define the sector occupancy time specific aircraft in a specific sector as the difference of the exit event time and the entry event time for that aircraft at that sector. The first analysis will show the accuracy of each tool in predicting events dependant on initial conditions such as imposed ground or air delay; this second analysis will show the accuracy of each tool in predicting events independent of these initial conditions. Indeed, sector occupancy time prediction is independent of conditions that occur before a predicted event that might introduce error to *both* exit and entry time predictions.

As with other studies in this document, we present the results of these as an aggregate of all flights and events over a certain time period. In this way, we compare the net ability of each tool to make predictions. Furthermore, it is often helpful to split presentation of the data into different types of flights (active or proposed, for example), into specific types of sectors (high or low, arrival or overflight) to gain insight into the strengths and weaknesses of tools. We will make these distinctions at the appropriate point in the discussion.

3.1.1 Sector Entry Time Analysis

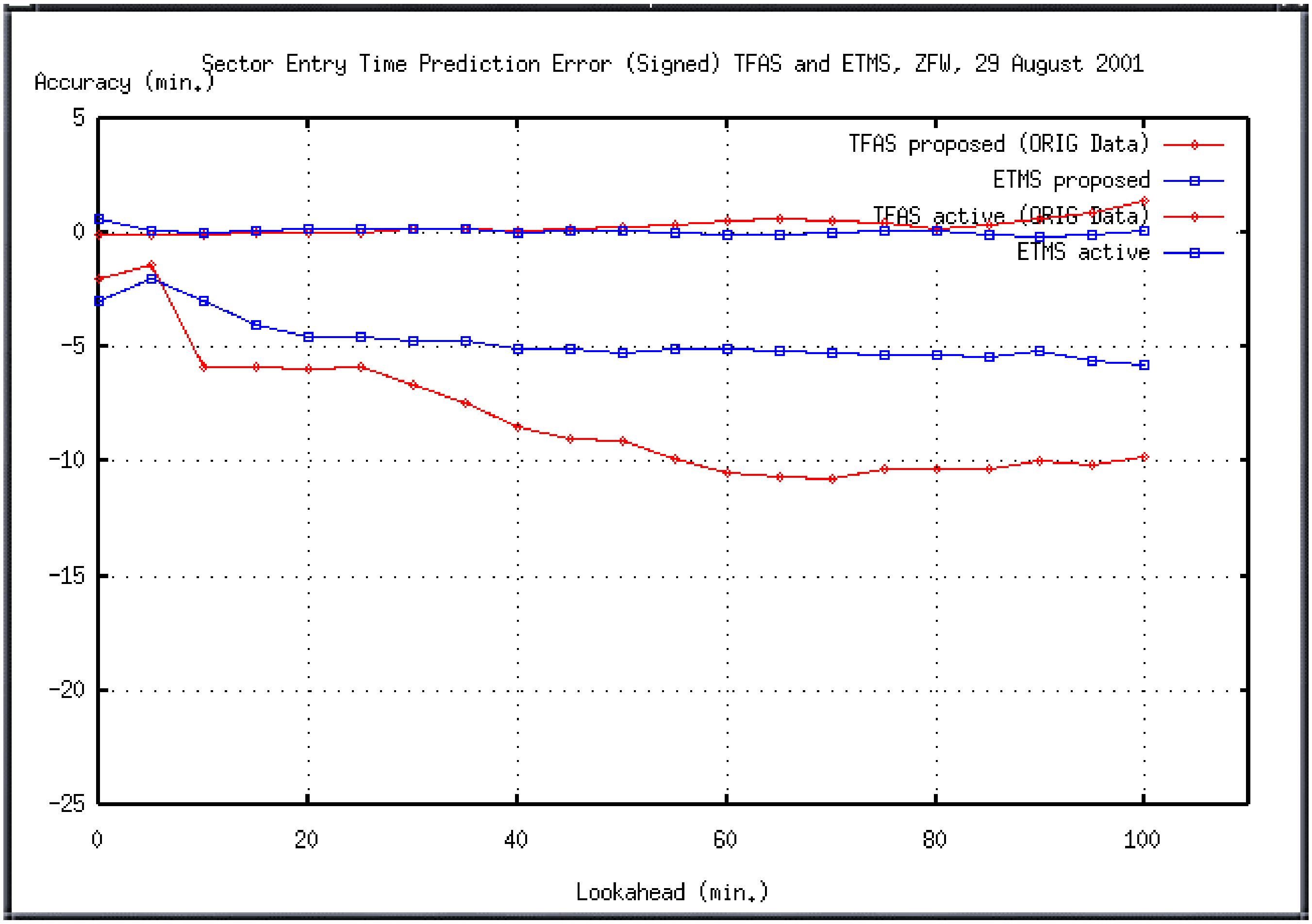
The purpose of this analysis is to compare the predictions of sector entry events made by TFAS and ETMS with the actual sector entry times, and thereby show the relative error in the predictive abilities of each. We used three primary data sources, namely the TFAS and ETMS sector entry and exit time predictions, and the actual sector entry and exit times for this analysis. We also used a fourth, secondary data source, which provides a list of aircraft ids to filter from the analysis. This list is a product of the CSEP program described in the previous section, where certain flights are filtered out for various reasons including insufficient raw data or user specified criteria.

For a specific event, there exists only one instance of it in the data set representing actual data. In each predicted dataset, however, multiple predictions for each aircraft are made. In the case of the TFAS dataset, one prediction is made for each flight plan change or track hit. For the ETMS dataset, predictions output every five minutes, with each new event list file. We performed a comparison between actual sector entry times and predicted sector entry times once each for TFAS and ETMS in the following manner:

1. For each instance of an actual aircraft/entry event pair, search the prediction data for sets of predictions made for that aircraft before the actual event occurred.
2. When such a prediction set is found, search the prediction set for the event referenced by the actual data.
3. If the event is found, record the discrepancy between the time at which the event actually occurred (from actual data) and the time in which the event was predicted to occur. This is referred to as the “prediction error”. Also record the difference between the time the event actually occurred and the time at which the prediction was made. This is the “time to prediction”.
4. Delete any prediction error/time to prediction pairs in which the prediction error is greater than 12 hours (or less than –12 hours), as these indicate events occurring for the same flight number on different days.
5. Delete any aircraft whose id is specified in the “reject_acids” file.
6. If the “intersection_only” option is passed to the program, reject any flights that do not appear in both TFAS and ETMS sources.
7. Group the individual prediction error/time to prediction values into a format for plotting. For all of the plots in this study, pairs were grouped into 5 minute buckets based on time to prediction; all pairs which describe a time to prediction of between 0 and 4 minutes are grouped into the first bucket, 5 to 9 in the next bucket, and so on.
8. For each bucket, filter outliers. For the purpose of this analysis, outliers are defined as anything occurring outside of 3 standard deviations of the set represented by the bucket.
9. For each bucket, take the average of the set and use the result for plotting. On the x axis of the plot is “time to prediction” for each bucket, on the y axis is “average error” for each bucket, or the result as described in the previous sentence. This is the signed plot of sector entry error. Also record the standard deviation of each bucket for plotting.
10. For each bucket, take the average of the absolute value of the set and use the result for plotting. On the x axis of the plot is “time to prediction” for each bucket, on the y axis is “average absolute error” for each bucket, or the result as described in the previous sentence. This is the absolute plot of sector entry error.

Notes: For step 1, the last exit event in an event list is treated as an entry event if the event represents the exit from the center. All other exit events are, of course, also entry events and handled accordingly. Step 3 indicates what to do when an event prediction is found. The case where an event prediction is not found is handled in a later analysis, namely that of “sector hit rate”. Steps 1 – 4 are handled by an analysis program “Event Compare”. Steps 5 – 8 are handled using simple “Octave” scripts. Both programs are discussed in the appendices. For some analyses, it was desirable to show only events at certain sectors (arrival sectors only, for instance). These cases are noted in the results.

This process results in plots of the signed sector entry time prediction error and the absolute sector entry time prediction error. The former is a good indicator whether a prediction tool is in general predicting flights to move too slowly or quickly. The latter is an indicator of absolute accuracy of the predictions made.



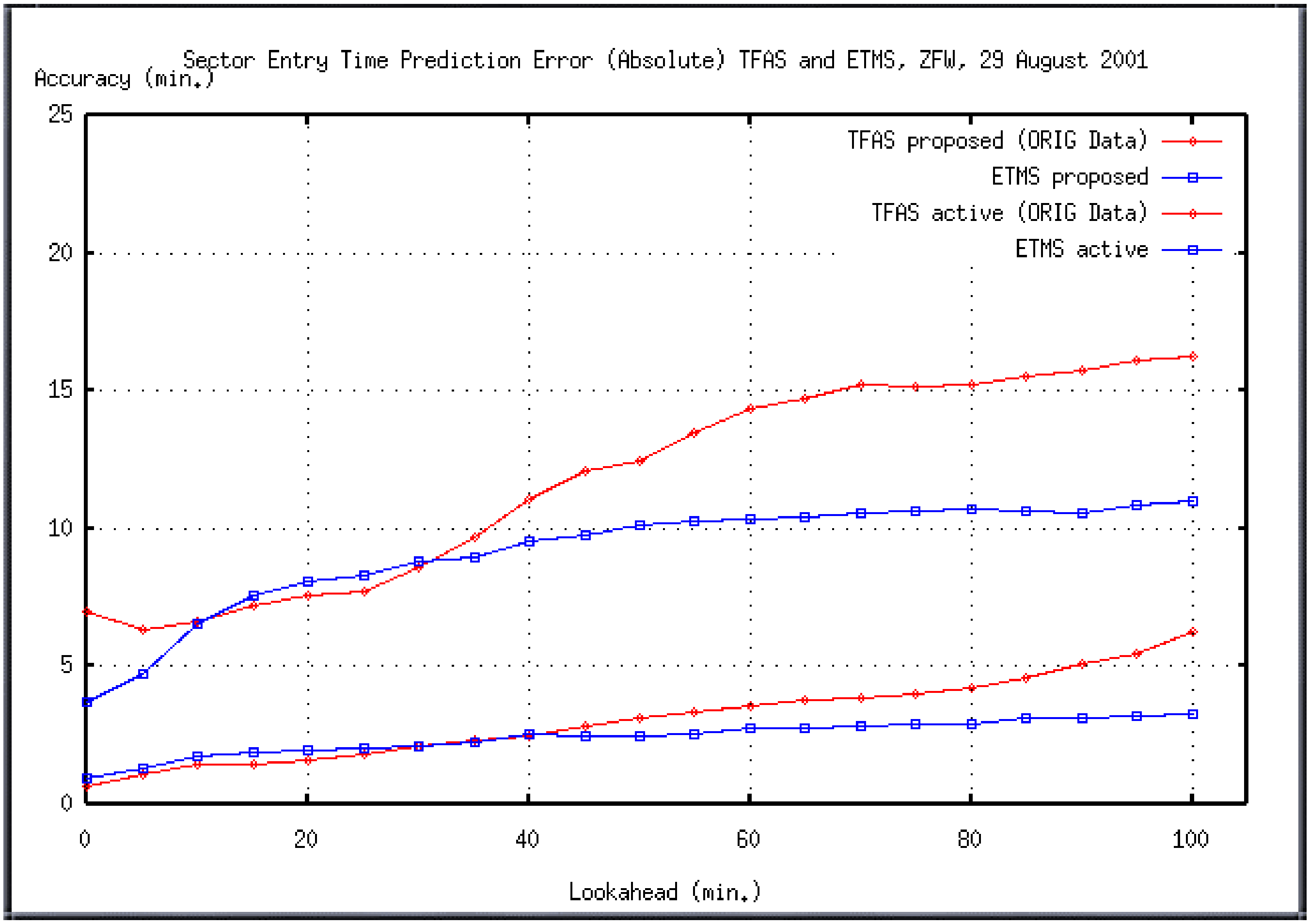
This plot shows the average error (in five minute increments) for TFAS and ETMS in predicting sector entry time for the August 29 data sample. For active flights, both TFAS and ETMS are relatively close to zero. This indicates that neither tool is predicting active flights to fly “slow” or “fast”, which is as expected. Past 60 minutes look-ahead, the plot shows that TFAS tends to predict sector entry times to be slightly later than actual.

For proposed flights, we notice a bias towards predicting flights to enter sectors earlier than they actually do in both tools. This is due to the one-sided nature of delays; flights often depart later than scheduled, but almost never earlier. The effect is more pronounced in the TFAS plot.

Data sample characteristics:
ETMS:
- Number actual ac 4193 (acs which make appearance during time period)
- Number actual ac after filtering 3898
- Number predicted ac 4640 (acs which make appearance during time period)
- Number predicted ac after filtering 3554
- Number actual events 20040 (valid ac only)
- Number predicted events 92113 (valid ac only)

TFAS:
- Number actual ac 4193 (acs which make appearance during time period)

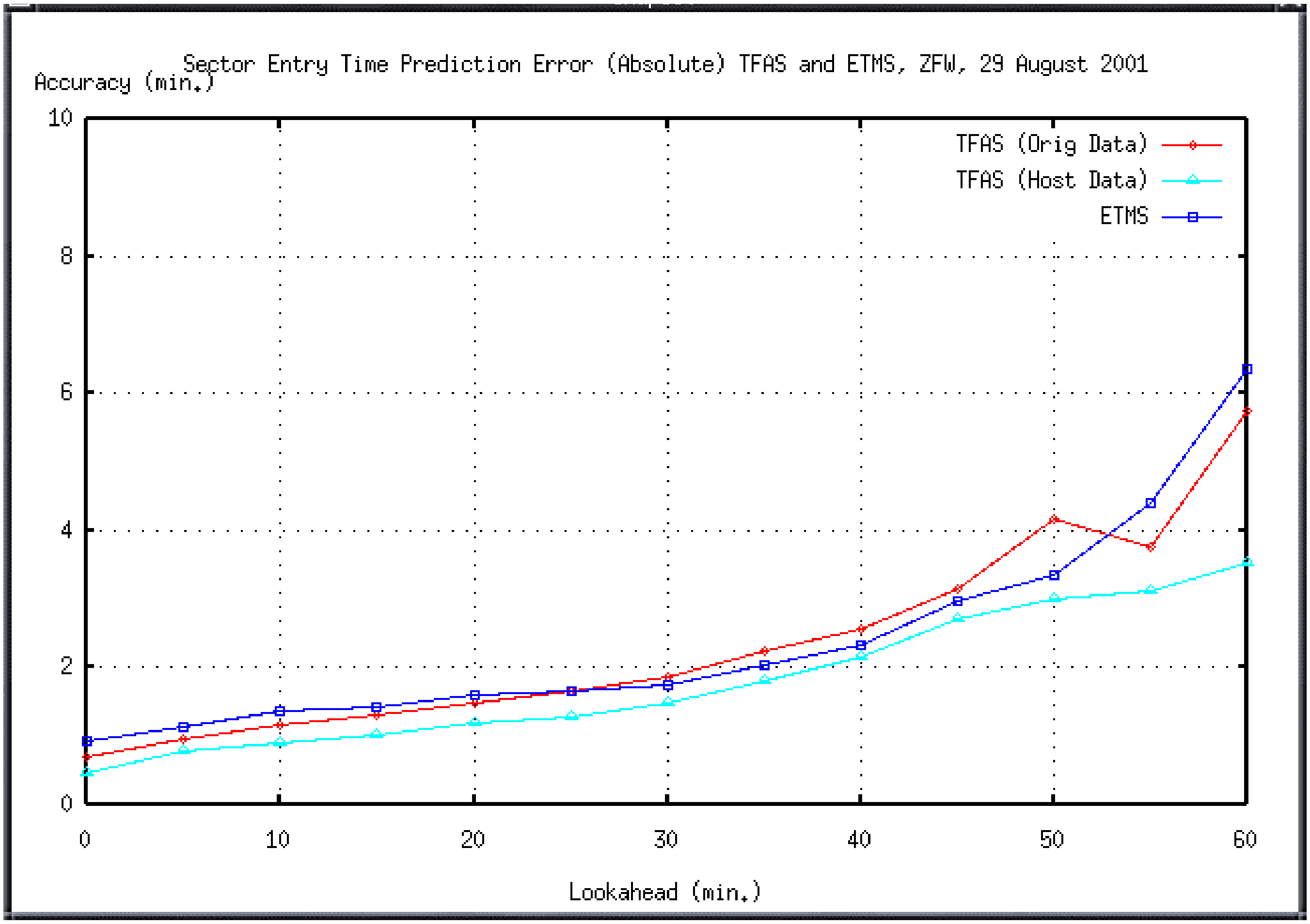
Number actual ac after filtering 3898
Number predicted ac 4525 (acs which make appearance during time period)
Number predicted ac after filtering 3439
Number actual events 20040 (valid ac only)
Number predicted events 325856 (valid ac only)



This plot shows the absolute entry time prediction error for TFAS and ETMS active and proposed flights. For both active and proposed, TFAS appears to perform slightly better for the first 35 minutes of look-ahead (except for the first 10 minutes for proposed flights. We expect the error for proposed flights to be higher in general, as proposed flights can be restricted with an arbitrarily large amount of ground delay, whereas active flights can not.

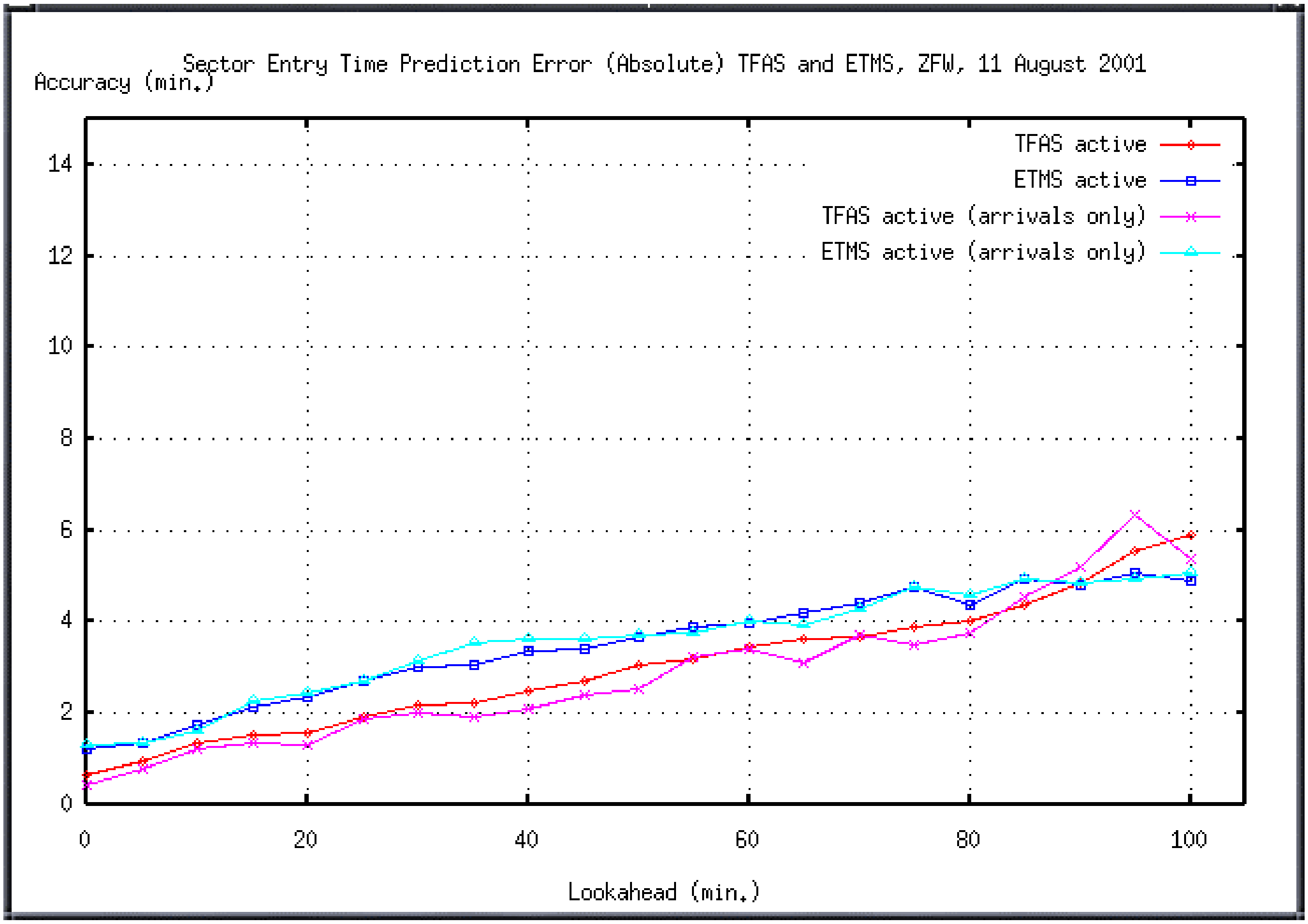
TFAS appears to perform poorly after 80 minutes of look-ahead. For all of our analyses, TFAS was running for ZFW in single center mode. In this state, the tool does not posses detailed adaptation information for surrounding centers, resulting in reduced prediction accuracy for flights in those centers. The effect is more pronounced for longer lookahead, as it represents more flights out of the center.

Data Sample Characteristics
(Same as previous)



This plot shows the absolute entry time prediction error for TFAS and ETMS active flights inside the center (ZFW), TFAS with both Orig data and Host data, for the August 29 sample. While predictions made by TFAS using Orig data become worse than ETMS around 30 minutes lookahead as we have seen before, TFAS using Host data appears to perform better than ETMS for the duration shown. It should be noted that all three plots degrade severely after about 45 minutes. This plot shows only active flights within the center; flights that remain in the center longer than 50 minutes characteristically fly either unusually slow, or in a meandering path. Both of these flight paths are associated with poor trajectory predictions.

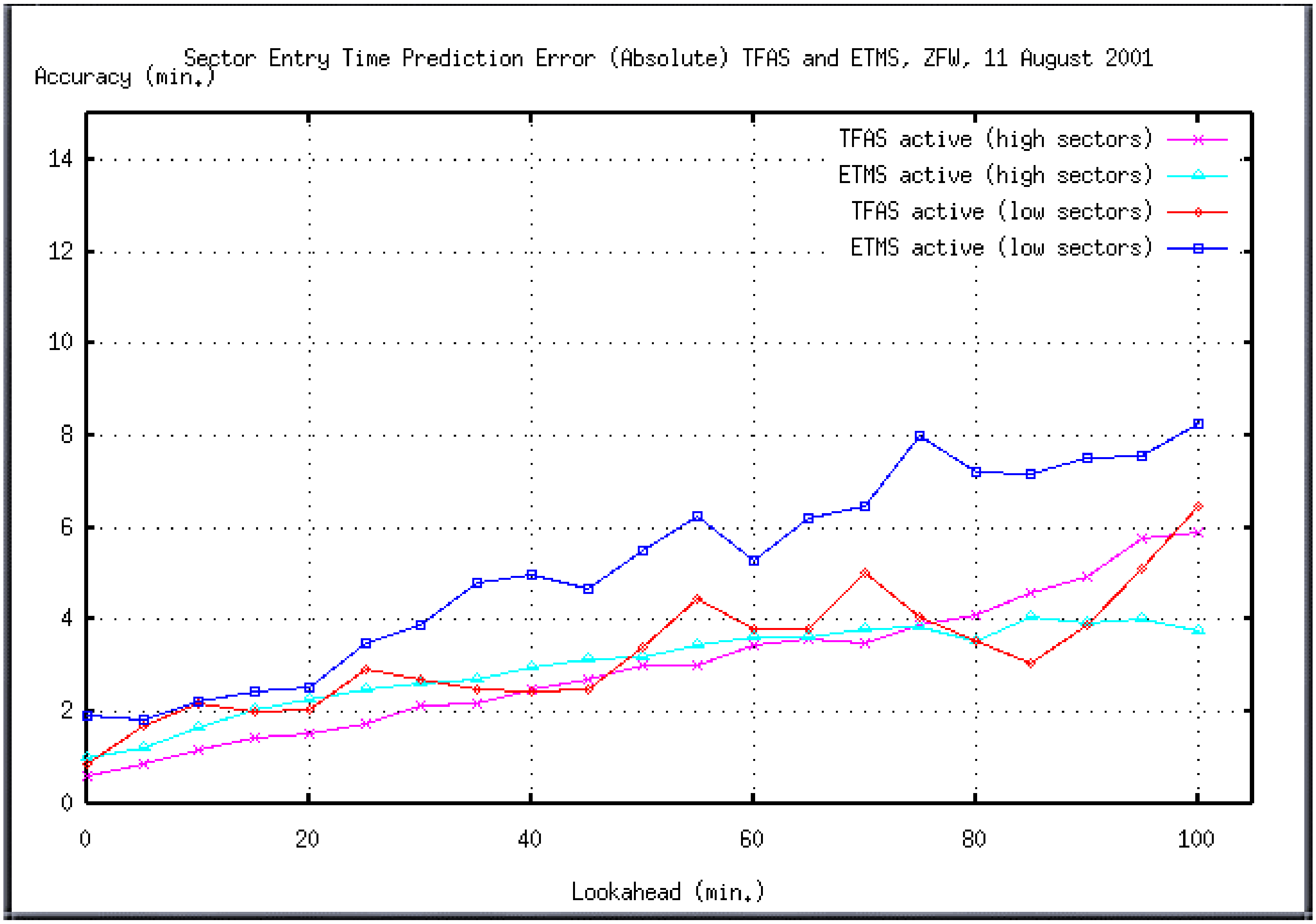
Data Sample Characteristics
(Same as previous)



Here we see the absolute entry time prediction error for TFAS and ETMS active aircraft, all flights vs arrivals only, for the August 11 sample. On this day, TFAS appears to perform significantly better than ETMS for the first 90 minutes lookahead for all flights and for arrivals only. TFAS performs slightly better when predicting arrivals than when predicting other flights. This is expected, as CTAS was developed primarily as an arrival tool. ETMS appears to perform nearly the same when predicting arrivals vs. all flights.

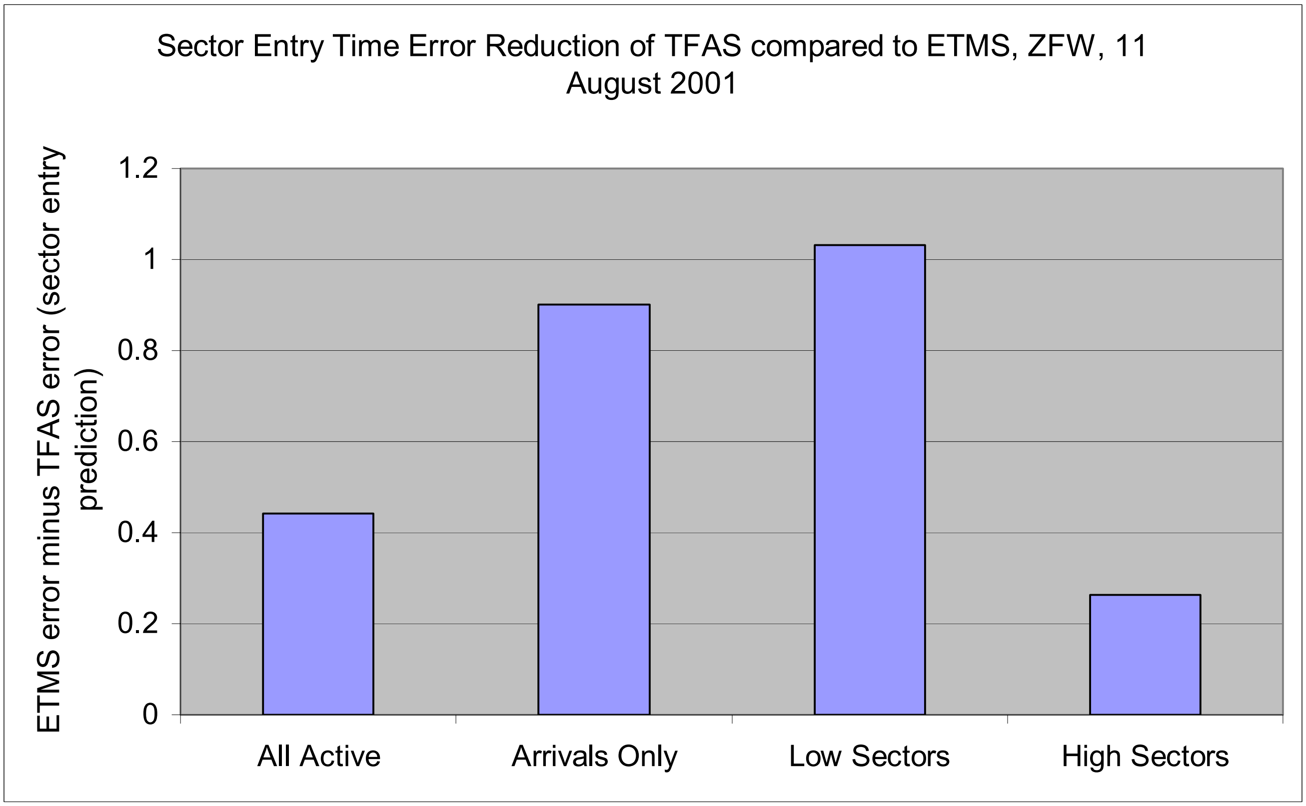
Data Sample Characteristics
ETMS
    Number actual ac 2237 (acs which make appearance during time period)
    Number actual ac after filtering 2190
    Number predicted ac 2453 (acs which make appearance during time period)
    Number predicted ac after filtering 2148
    Number actual events 11440 (valid ac only)
    Number predicted events 40449 (valid ac only)
TFAS
    Number actual ac 2237 (acs which make appearance during time period)
    Number actual ac after filtering 2190
    Number predicted ac 2380 (acs which make appearance during time period)

Number predicted ac after filtering 2075
Number actual events 11440 (valid ac only)
Number predicted events 191568 (valid ac only)



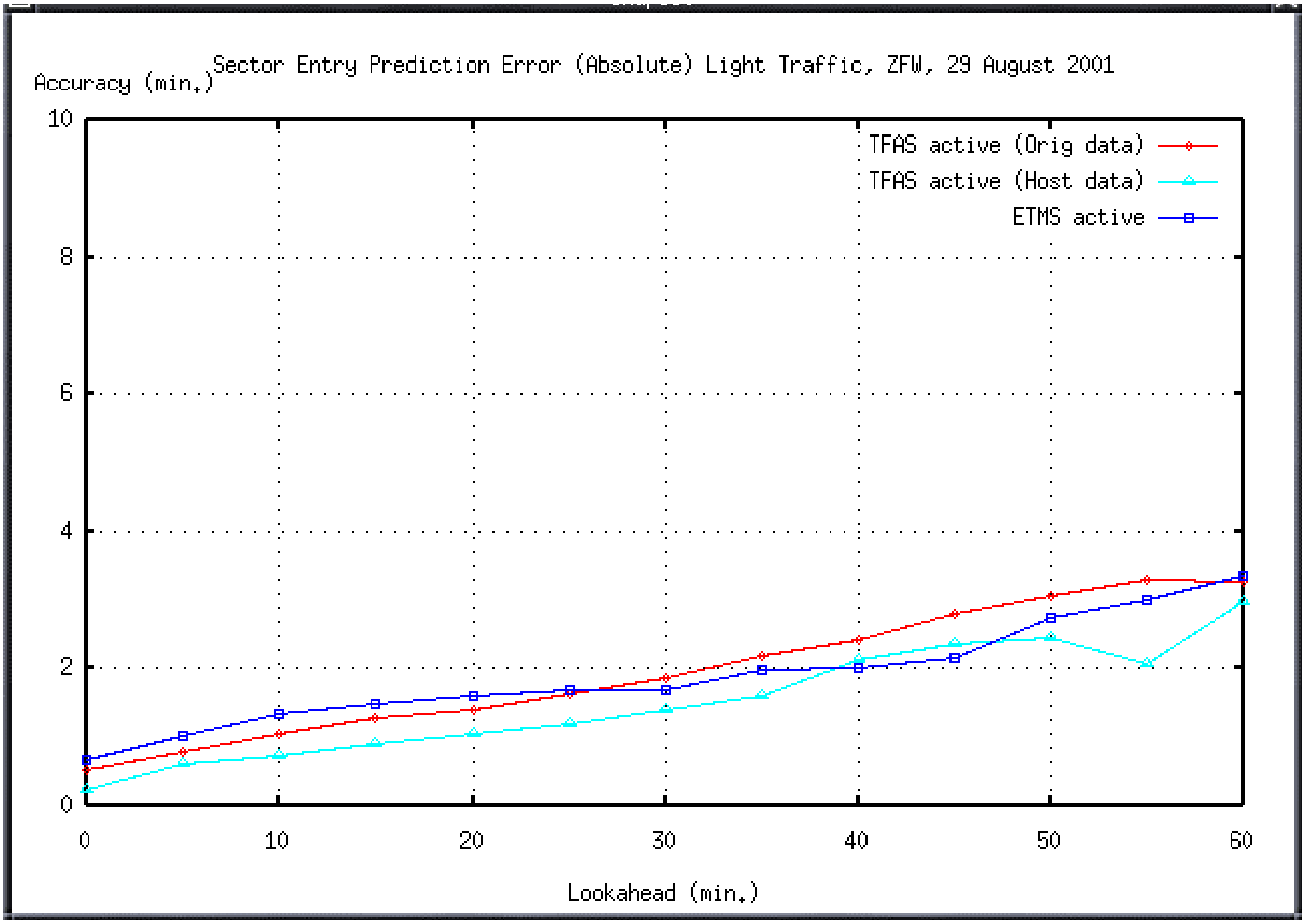
This plot shows the absolute entry time prediction error for TFAS and ETMS active aircraft for low and high sectors for the August 11 sample. For high sectors, TFAS performs slightly better than ETMS for about the first 75 minutes lookahead. For low sectors, TFAS performs better than ETMS for the duration of the analysis. We expect the characteristic rough plots for lower sectors as those sectors tend to be associated with general aviation aircraft for which it is difficult to make trajectory predictions.

Data Sample Characteristics
(Same as previous)



This plot shows the error reduction of TFAS compared to ETMS for the August 11 sample. As we have seen in other graphs, TFAS shows greatest improvement over ETMS for arrivals and in low sectors.

Data Sample Characteristics
(Same as previous)



This plot shows the entry time prediction error for TFAS and ETMS during a period of light traffic for the August 29 sample. It is important in that it minimizes the effects of delay and rerouting associated with high capacity, and better portrays the characteristics of the trajectory engine of each tool. In order to further isolate this element, this plot is of high sectors only.

This plot exhibits much of the same characteristics of those that capture all sectors over the entire day; TFAS using orig data tends to perform better than ETMS for the first 25-30 minute lookahead. However, in this sample, TFAS using Host data performs better than ETMS for almost the entire duration.

Data Sample Characteristics

Time period from 0000Z to 0600Z
ETMS
Number actual ac 993 (acs which make appearance during time period)
Number actual ac after filtering 944
Number predicted ac 1129 (acs which make appearance during time period)
Number predicted ac after filtering 864
Number actual events 4640 (valid ac only)
Number predicted events 14744 (valid ac only)

TFAS (Orig data)

  Number actual ac 1286 (acs which make appearance during time period)
  Number actual ac after filtering 1216
  Number predicted ac 1363 (acs which make appearance during time period)
  Number predicted ac after filtering 140
  Number actual events 6004 (valid ac only)
  Number predicted events 70189 (valid ac only)

TFAS (Host data)

  Number actual ac 993 (acs which make appearance during time period)
  Number actual ac after filtering 944
  Number predicted ac 1065 (acs which make appearance during time period)
  Number predicted ac after filtering 800
  Number actual events 4640 (valid ac only)
  Number predicted events 182811 (valid ac only)

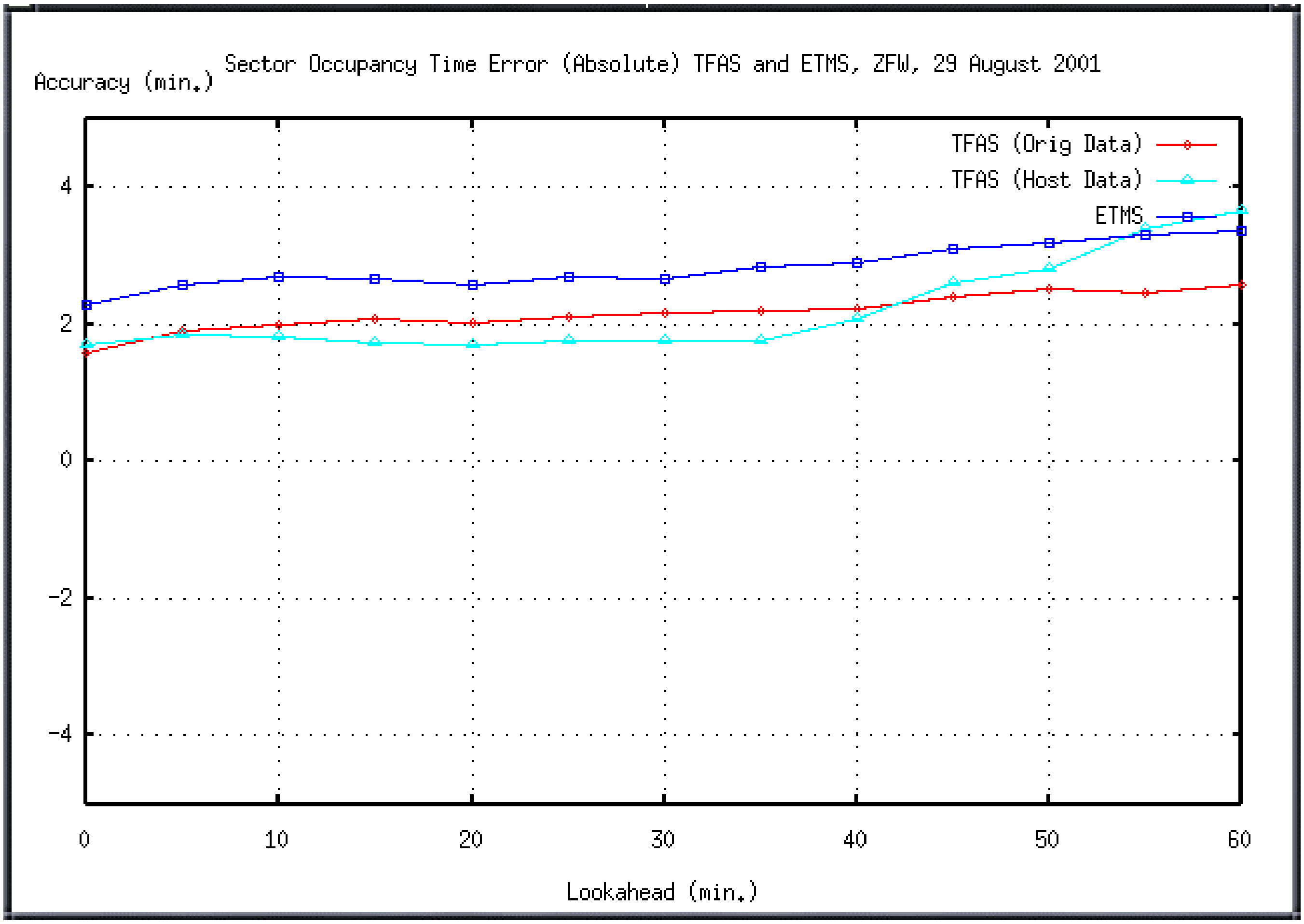
Sector Occupancy (Duration) Analysis

The purpose of this analysis is to compare sector occupancy time predictions made by TFAS and ETMS with the actual sector occupancy time. The occupancy time is defined difference in time for an aircraft entry event into a sector and the aircraft exit event out of the same sector. An aircraft may be delayed (or circled) in flight, or may be rerouted for weather. These unforeseeable events affect the entry time predictions for all subsequent events. An analysis more independent of these cases is that of sector occupancy.

The process to obtain this data is almost exactly the same as that used in the previous section to obtain sector entry time errors. The only difference is the data recorded in step 3 and the results obtained in steps 7 and 8. These are described as follows:

3. If the event is found, record the discrepancy between the actual sector occupancy time (exit time – entry time) and the predicted sector occupancy time. This is referred to as the “prediction error”. Also record the difference between the time the entry event actually occurred and the time at which the prediction was made. This is the “time to prediction”.
7. For each bucket, take the average of the set and use the result for plotting. On the x axis of the plot is “time to prediction” for each bucket, on the y axis is “average error” for each bucket, or the result as described in the previous sentence. This is the signed plot of sector occupancy error. Also record the standard deviation of each bucket for plotting.
8. For each bucket, take the average of the absolute value of the set and use the result for plotting. On the x axis of the plot is “time to prediction” for each bucket, on the y axis is “average absolute error” for each bucket, or the result as described in the previous sentence. This is the absolute plot of sector occupancy error.

This process results in plots of the signed sector occupancy time prediction error and the absolute sector occupancy time prediction error. The former is an indicator whether a prediction tool is in general predicting flights to move too slowly or quickly, relatively independent of unforeseeable future conditions. The latter is an indicator of absolute accuracy of the predictions made, relatively independent of unforeseeable future conditions. Both of these claims are qualified by saying “relatively independent”, because there is always a chance that an unforeseeable future condition (like imposed delay) might occur *in* one of the sectors in question, thereby affecting the actual occupancy time for that sector.



This plot shows the absolute error in predicting sector occupancy time for TFAS and ETMS, TFAS using both Orig and Host data, for the August 29 sample. All three plots show a significant amount of error in predicting the occupancy time of aircraft, suggesting that there is room for improvement in trajectory prediction. For this sample, TFAS using Orig data appears to predict on average about 0.5 minutes better than ETMS. While TFAS using Host data appears to predict better still, the plot degrades after about 40 minutes. This is the result of the radar horizon for host data occurring just outside of the center, whereas orig data includes the entire CONUS. There are a limited number of flights that remain in the center for longer than 40 minutes, and those that do tend to be associated with difficult trajectory predictions (meandering flight paths, etc). The effect is not as pronounced in ETMS and TFAS with Orig data, as the data includes predictions with longer than 40 minute lookahead made for active flights outside the center.

Data Sample Characteristics
ETMS:

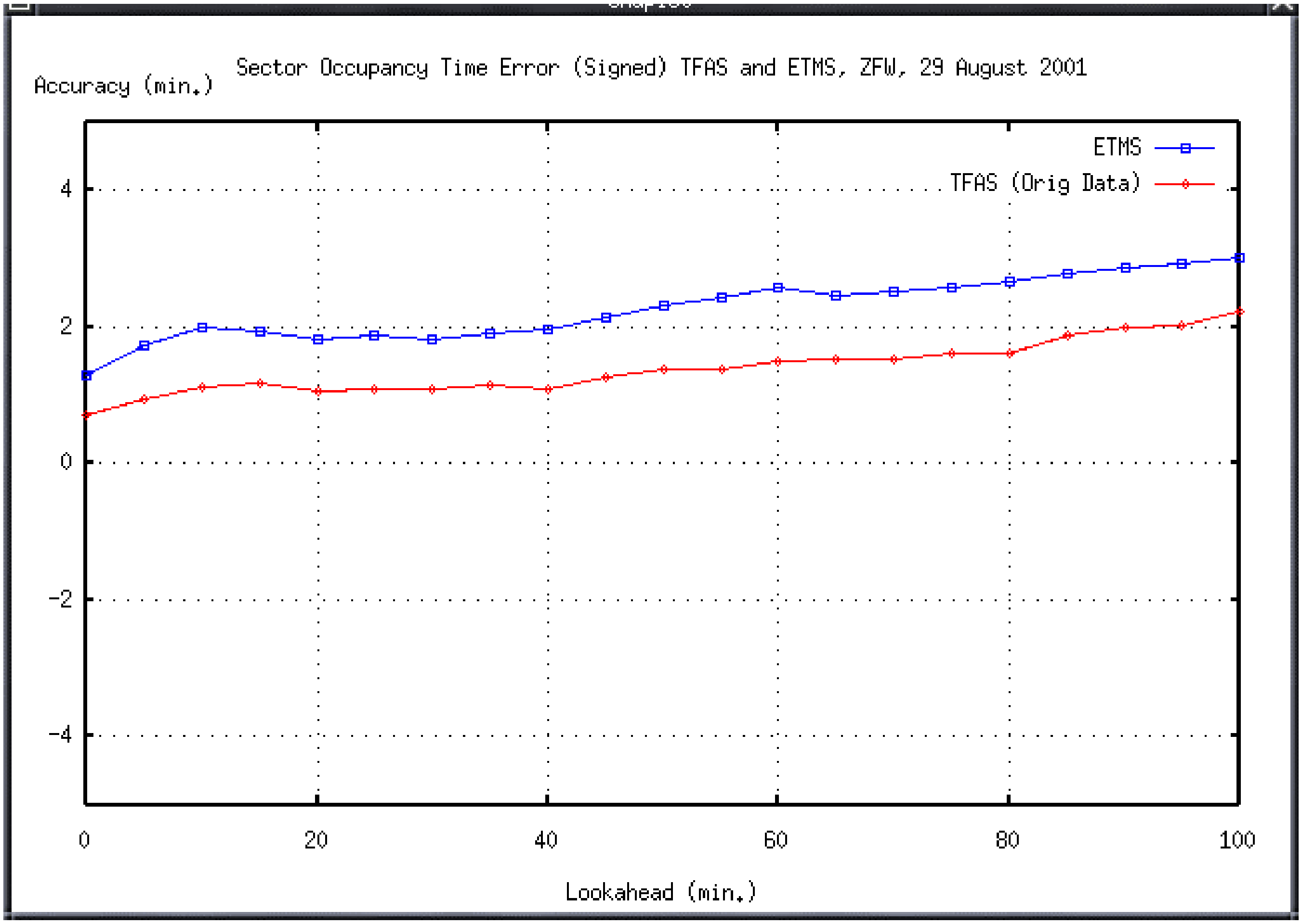
Number actual ac 4193 (acs which make appearance during time period)
Number actual ac after filtering 3898
Number predicted ac 4640 (acs which make appearance during time period)
Number predicted ac after filtering 3554
Number actual events 20040 (valid ac only)
Number predicted events 92113 (valid ac only)

TFAS (Orig):

- Number actual ac 4193 (acs which make appearance during time period)
- Number actual ac after filtering 3898
- Number predicted ac 4525 (acs which make appearance during time period)
- Number predicted ac after filtering 3439
- Number actual events 20040 (valid ac only)
- Number predicted events 325856 (valid ac only)

TFAS (Host):

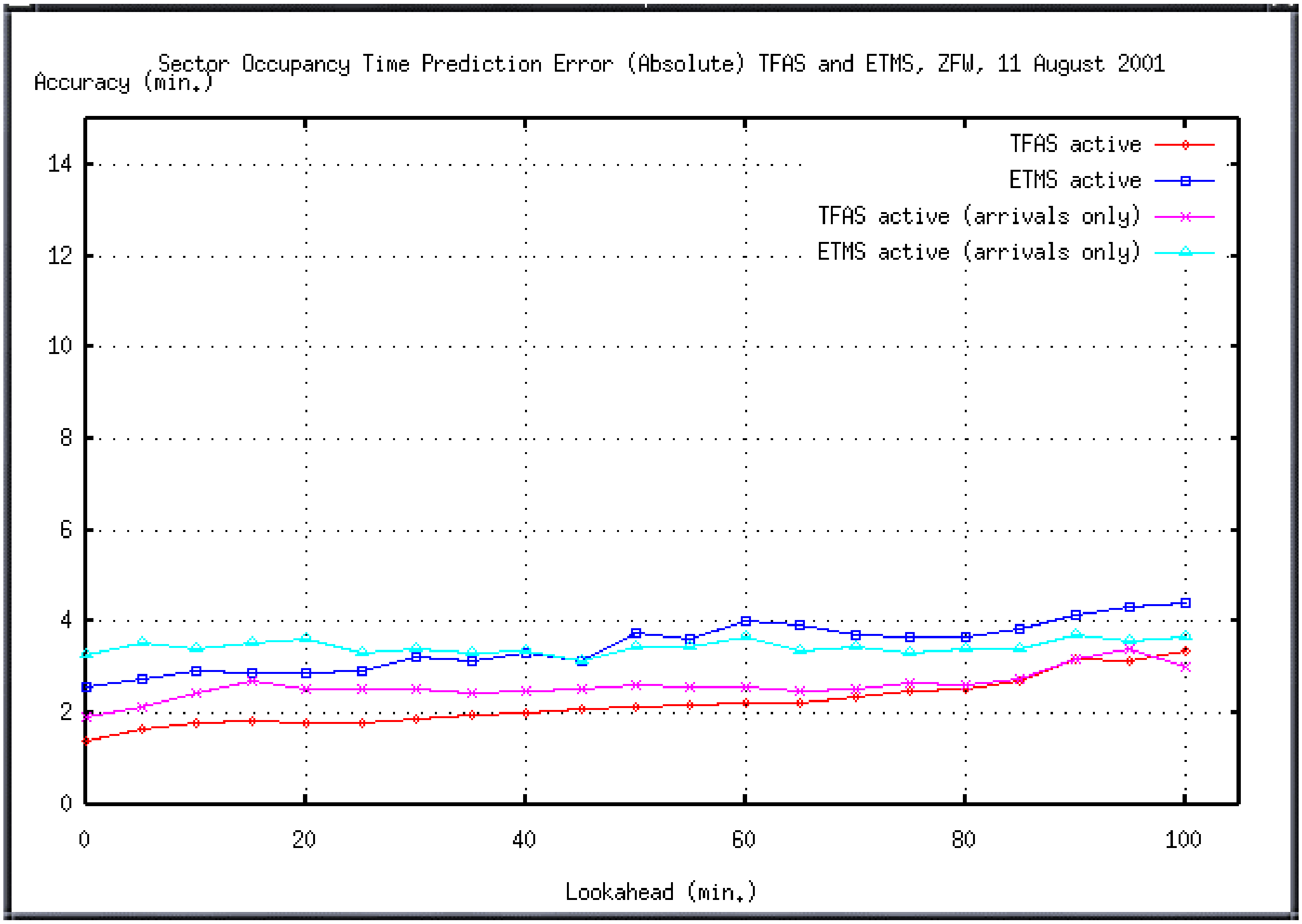
- Number actual ac 4657 (acs which make appearance during time period)
- Number actual ac after filtering 4322
- Number predicted ac 4902 (acs which make appearance during time period)
- Number predicted ac after filtering 3555
- Number actual events 22104 (valid ac only)
- Number predicted events 950312 (valid ac only)



This plot show the signed error in predicting sector occupancy time for TFAS and ETMS for the August 29 sample.

Data Sample Characteristics:

(Same as previous)



Here we see the sector occupancy time prediction absolute error for TFAS and ETMS, both for all flights and arrivals only, for the August 11 sample. Here TFAS performs better than ETMS in both cases. It is interesting to note that for shorter lookahead, both tools tend to do worse for arrivals than over the average, but for longer lookhead TFAS predictions tend to converge and ETMS actually improves slightly.

Data Sample Characteristics

ETMS

Number actual ac 2237 (acs which make appearance during time period)
Number actual ac after filtering 2190
Number predicted ac 2453 (acs which make appearance during time period)
Number predicted ac after filtering 2148
Number actual events 11440 (valid ac only)
Number predicted events 40449 (valid ac only)

TFAS

Number actual ac 2237 (acs which make appearance during time period)
Number actual ac after filtering 2190
Number predicted ac 2380 (acs which make appearance during time period)
Number predicted ac after filtering 2075
Number actual events 11440 (valid ac only)
Number predicted events 191568 (valid ac only)

3.2 Sector Demand Prediction Accuracy

While contingent upon trajectory predictions, sector demand accuracy is generally more useful to center controllers and TMU's. Predicted sector counts allow these users to mitigate the flow of traffic to prevent future airspace overloads and possibly limit unnecessary airspace and ground restrictions.

Sector count prediction is the ability of a tool to predict for any sector the number of aircraft to be in that sector at some time in the future. The first study analysis of this section will concern simply the accuracy of TFAS and ETMS in predicting sector counts. While this is a good indicator of the usefulness of either tool, it fails to address the accuracy of how individual flights contribute to those counts. For example, an aircraft may be predicted to occupy a sector that it never actually enters. Another aircraft, *not* predicted to enter the sector does in fact enter it at around the same time the first aircraft was predicted to do so. The sector count prediction for this particular sector would have been correct, although two aircraft actually occupied unpredicted sectors.

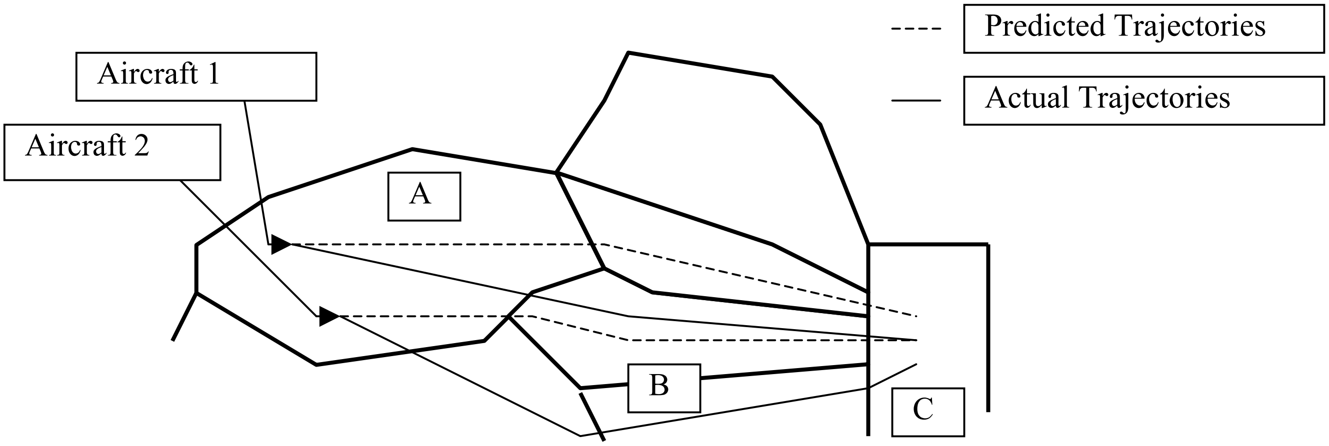


Figure 3 Predicted Aircraft 1 contributes to the count of sector B.

In this case, even though both aircraft suffered from poor predictions, the overall sector count prediction was correct. This effect over the aggregate is significant.

A second analysis, then, is presented in which the sector hit rate of each tool is compared. This is a comparison of the accuracy of each tool in simply predicting that an aircraft entered a sector at all. While this may not be significant in a tool that portrays predictions in the aggregate, it is vital to the operation of a tool depicting unique flights.

The results of these analyses are presented as an aggregate of all flights and events over a certain time period. In this way, the net ability of each tool to make predictions can be compared directly. Furthermore, it is often helpful to split presentation of the data into different types of flights (active or proposed, for example), into specific types of sectors (high or low, arrival or overflight) to gain insight into the strengths and weaknesses of tools. Those distinctions will be made at the appropriate point in the discussion.

3.2.1 Sector Count Accuracy Analysis

The purpose of this analysis is to compare the accuracy of TFAS and ETMS in predicting sector counts. For any given sector and time period, the sector count refers to the maximum count of aircraft in the sector at any instantaneous time. Three primary data sources, namely the TFAS and ETMS sector entry and exit time predictions, and the actual sector entry and exit times are used for this analysis. A fourth, secondary data source, provides a list of aircraft ids to filter from the analysis. This list is generally a product of the CSEP program described in the previous section, where certain flights are filtered out for various reasons including insufficient raw data or user specified criteria.

The data sources contain information about aircraft event lists, that is the path of an aircraft through sectors. The process of processing aircraft event lists into sector count is done in the following way:

1. Delete any aircraft that appear in the “reject_acids” file.
2. Associate with every sector one minute occupancy buckets that start at the beginning of the time period of the analysis and end at the end of the time period of the analysis.
3. For actual events, step through the event lists of each aircraft and, based on sector entry and exit times, add them to the appropriate one minute sector occupancy buckets (do this inclusively; if a flight enters a sector 40 seconds after the start of the minute interval, it is put into the bucket for that minute).
4. For predicted events, step through the time period of the analysis in one minute increments. For each aircraft, locate the prediction made at a time closest to but before the current time increment. Step through the event lists of each aircraft and, based on sector entry and exit times, add them to the appropriate 1 minute sector occupancy buckets. Note that there will be a different list of sectors with 1 minute occupancy buckets for EACH MINUTE in the time period.

Note: These steps are handled in this analysis by an analysis program “event compare”. The details of this programs are discussed in the appendices.

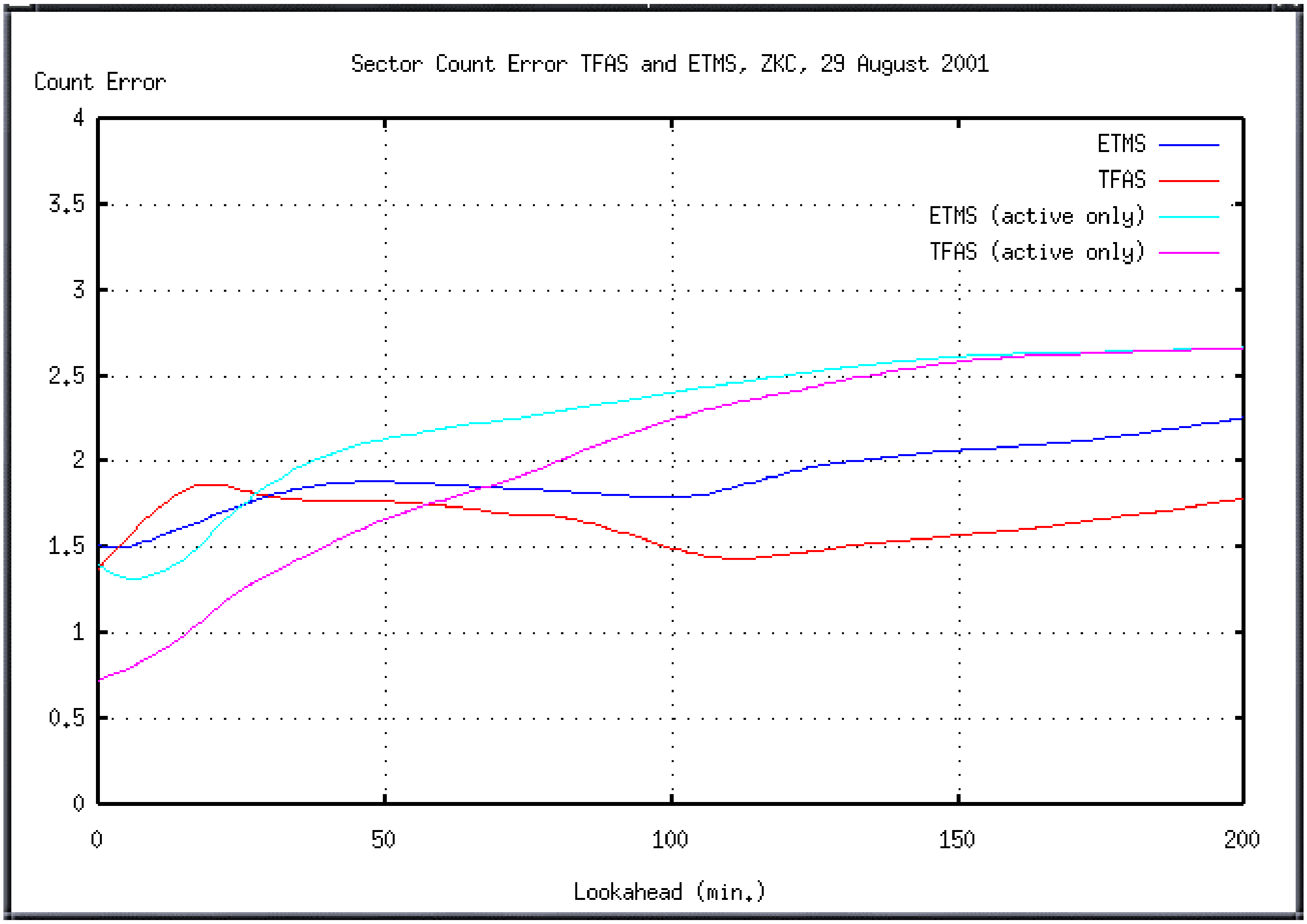
The result of this process is a set of actual sector counts in one minute buckets, and several sets of predicted sector counts (one for each minute of the time period). What remains to be done is compare the actual and predicted counts. This is as follows:

1. Step through the time period of the analysis in one minute increments. This will be the “current time”. At each increment, create a 10 minute “window” that will move through time in one minute increments, starting at the current time.
2. For each sector, compare the predicted maximum instantaneous count (maximum count over any of the 1 minute buckets) with the actual maximum instantaneous count. Record this difference as the “sector count error”. Associate with the sector count error the time difference between the beginning of the 10 minute window and the current time. This is the “time to prediction”
3. Group the individual prediction error/time to prediction values into a format for plotting. For all of the plots in this analysis, pairs were grouped into their respective 1 minute buckets, according to “time to prediction”.
4. For each bucket, filter outliers. For the purpose of this analysis, outliers are defined as anything occurring outside of 3 standard deviations of the set represented by the bucket.
5. For each bucket, take the average of the set and use the result for plotting. On the x axis of the plot is “time to prediction” for each bucket, on the y axis is “average error” for each bucket, or the result as described in the previous sentence. This is the signed plot of sector count error. Also record the standard deviation of each bucket for plotting.

6. For each bucket, take the average of the absolute value of the set and use the result for plotting. On the x axis of the plot is “time to prediction” for each bucket, on the y axis is “average absolute error” for each bucket, or the result as described in the previous sentence. This is the absolute plot of sector count error.

Note: Steps 1 and 2 in the process are handled by an analysis program “event compare”. Steps 3 – 6 are handled by simple “Octave” scripts. Both programs are detailed in the appendices.

This process results in plots of the signed sector occupancy time prediction error and the absolute sector occupancy time prediction error. The former is a good indicator whether a prediction tool is in general predicting sector counts too high or low. The latter is an indicator of absolute accuracy of the predictions made.



This plot shows the absolute sector count error for ETMS and TFAS, both showing all flights and active flights only. The thing to note about the TFAS plot showing all flights is that it exhibits large error in the first 20 minutes lookahead, slowly drops off to a low point around 110 minutes and then begins to ramp up. There is a similar effect in the ETMS data, though not as pronounced.

The “hump” at the left side of each plot is due to the error introduced by proposed flights near departure. As a later plot will show, this region is associated with overprediction, that is, predicting more flights in the sector than actually occupy the sector. These proposed flights get delayed and/or cancelled, and their effect is minimized past 100 minutes lookahead. Errors after that point are associated with underprediction. Plots of the sector count error associated with active flights only was included to illustrate this feature- they exhibit a gradual rise in error as lookahead time increases, as we would expect.

For this sample, TFAS tends to be associated with lower absolute error in predicting sector counts than ETMS.

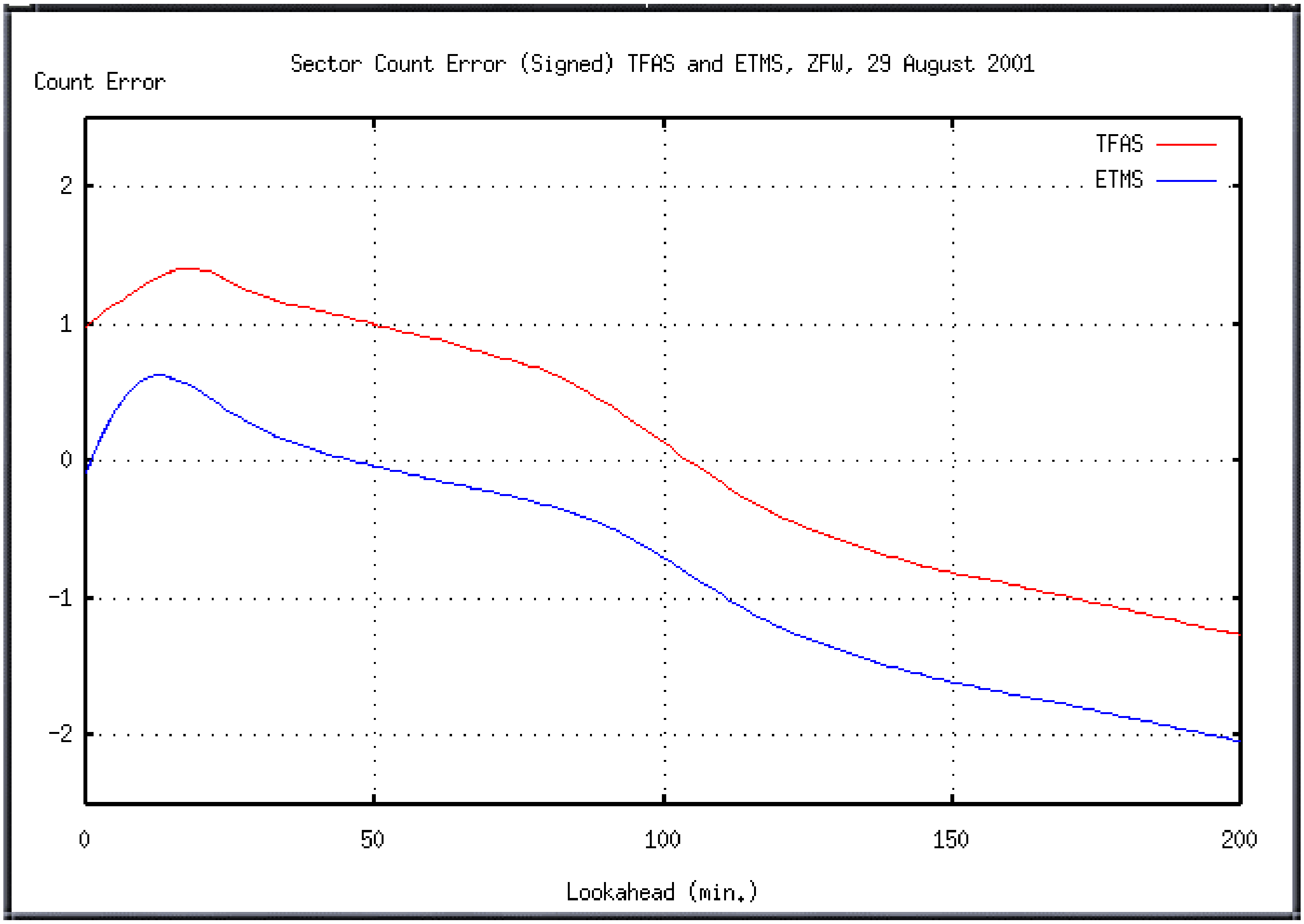
Data Sample Characteristics:
ETMS:

Number actual ac 4193 (acs which make appearance during time period)
Number actual ac after filtering 3898

Number predicted ac 4640 (acs which make appearance during time period)
Number predicted ac after filtering 3554
Number actual events 20040 (valid ac only)
Number predicted events 92113 (valid ac only)

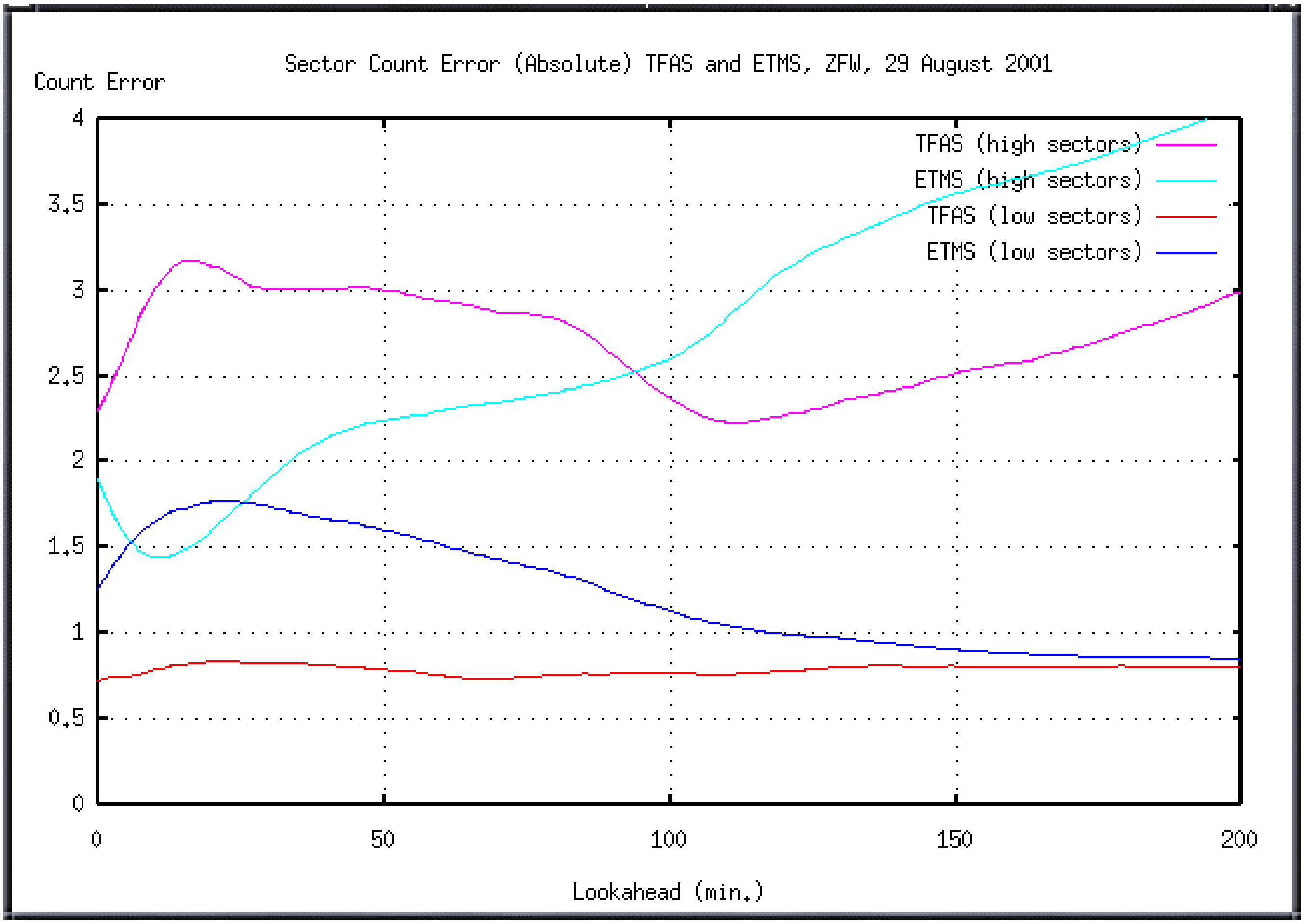
TFAS (Orig):

Number actual ac 4193 (acs which make appearance during time period)
Number actual ac after filtering 3898
Number predicted ac 4525 (acs which make appearance during time period)
Number predicted ac after filtering 3439
Number actual events 20040 (valid ac only)
Number predicted events 325856 (valid ac only)



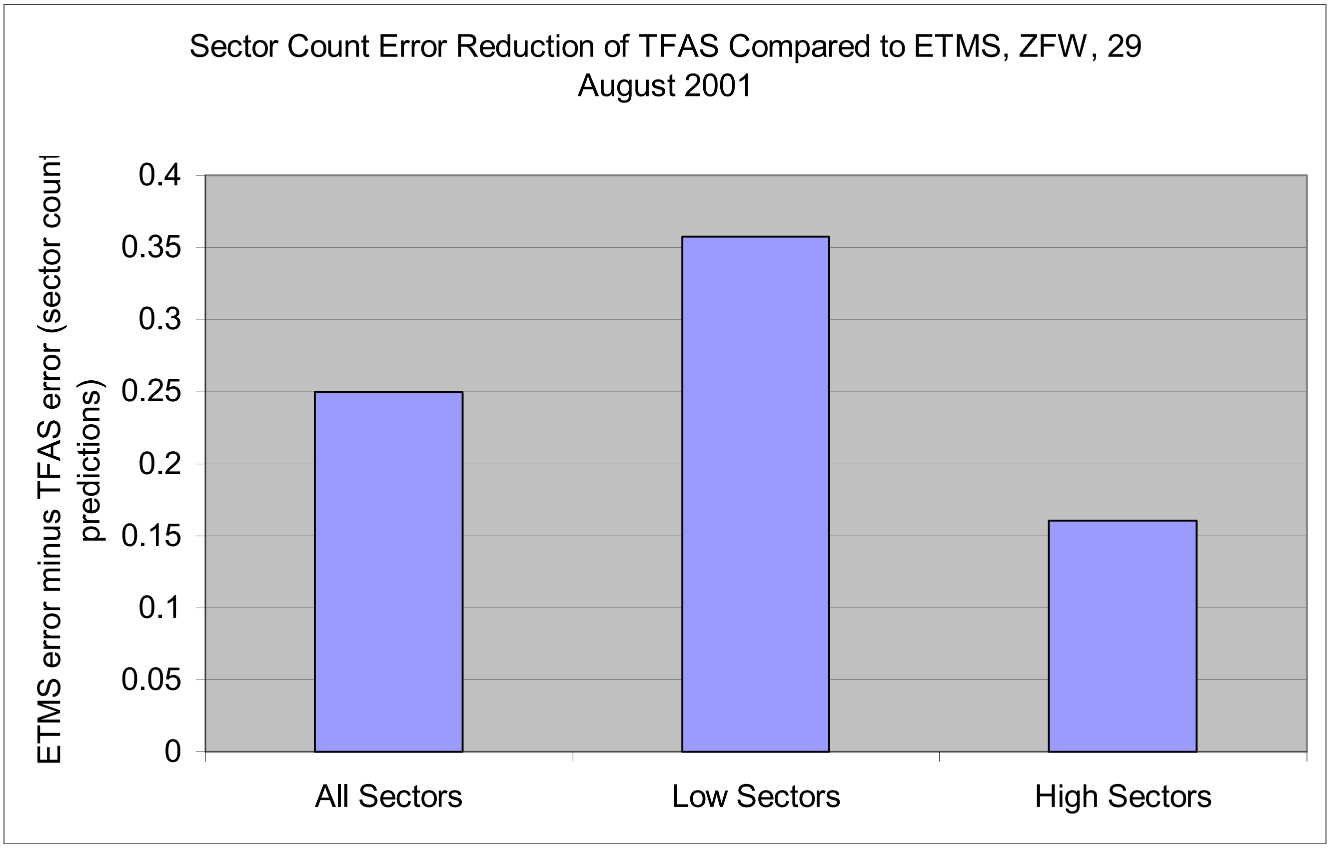
This plot shows the signed sector count error for TFAS and ETMS. It helps to illustrate the point made with the previous plot, that for small lookahead errors are associated with overprediction, and for large lookahead they are associated with overprediction. As stated before, this is due to proposed flights near departure and their associated delays and cancellations.

Data Sample Characteristics:
(Same as previous)



Here we see the absolute sector count error for TFAS and ETMS, both for low and high sectors, for the August 29 data sample. Again we see the effect of proposed aircraft for early lookahead, though the effect appears to be more pronounced in high sectors for TFAS and low sectors for ETMS, possibly suggesting different altitude profiles for departures. On this day, TFAS performs better than ETMS for low sectors, and for lookaheads longer than 90 minutes for high sectors.

Data Sample Characteristics
(Same as previous)



This graph shows the error reduction of TFAS in sector count prediction as compared to ETMS. For this day TFAS reduces the error in sector count prediction. The effect is more pronounced for low sectors.

Data Sample Characteristics
(Same as previous)

While this analysis is a good indicator of the behavior of a tool on the aggregate in predicting sector counts, it does not guarantee accuracy of sector occupancy for individual flights. The next section deals with an analysis of that, known as sector hit rate. Sector hit rate, together with sector count prediction indicate the overall accuracy of a tool in predicting flights to be in the correct sectors at the correct times.

3.2.2 Sector Hit Rate Analysis

The purpose of this analysis is to compare the accuracy of the contribution of predictions of individual aircraft to sector count predictions made by TFAS and ETMS. While sector count is not necessarily dependant on individual predictions, understanding the accuracy with which these individual predictions is made is crucial the confidence with which controllers use the tool.

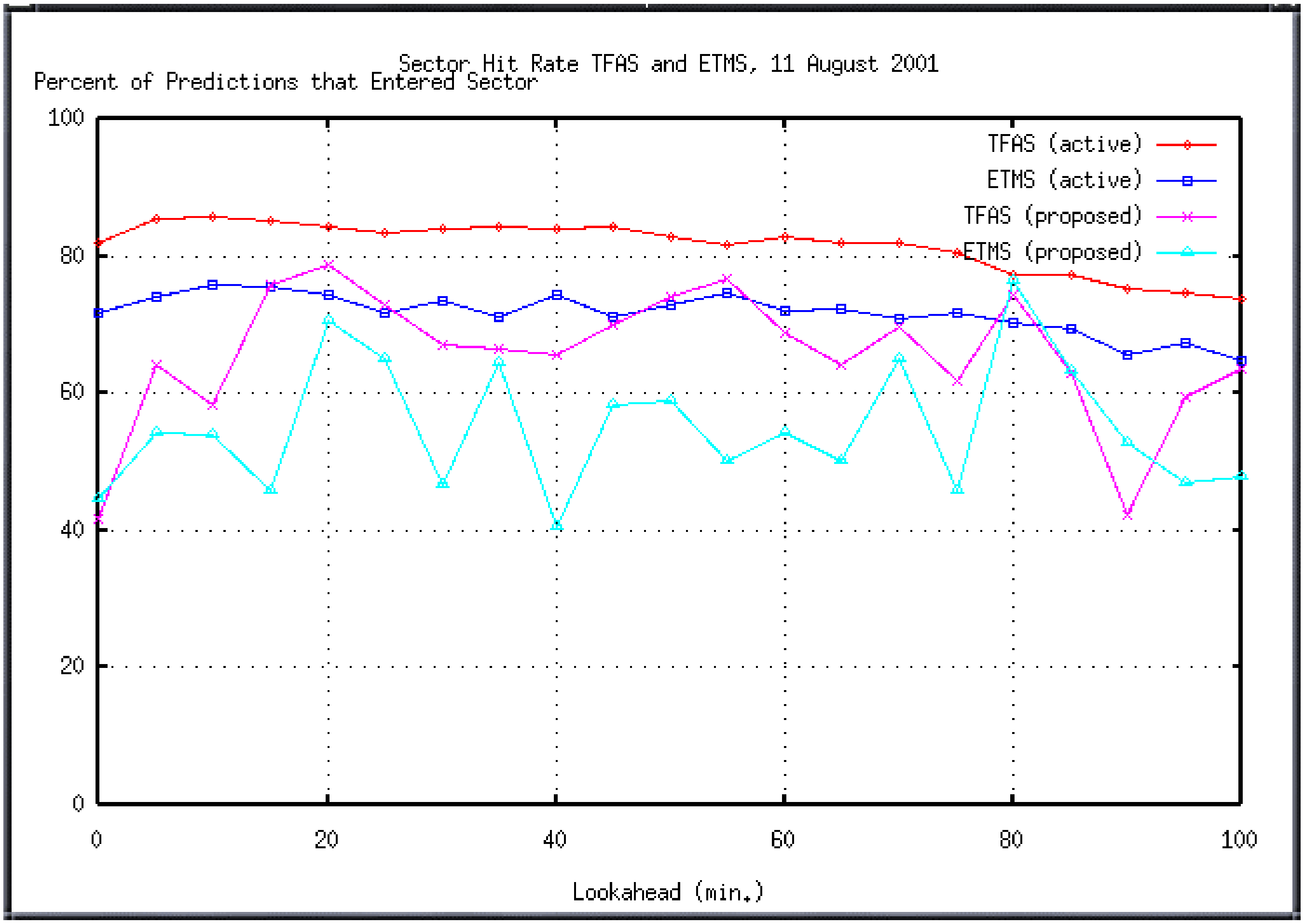
Sector hit rate refers to the success with which a tool predicts an aircraft to occupy a sector at some time in the future. This analysis is different than that of Sector Entry time prediction, since it compares not the accuracy in terms of time of an event, but rather the prediction of whether or not the aircraft will ever hit that event.

For a specific event, there exists only one instance of it in the data set representing actual data. In each predicted dataset, however, multiple predictions for each aircraft are made. In the case of the TFAS dataset, one prediction is made for each flight plan change or track hit. For the ETMS dataset, predictions are usually made every five minutes. A comparison between actual sector entry times and predicted sector entry times is therefore performed once each for TFAS and ETMS in the following manner:

1. For each instance of a predicted aircraft/entry event pair, search the actual data for that event (also note that the events must occur within one hour of each other.
2. If the event is found, record a "sector hit". Also record the difference between the time the event was predicted to occur and the time at which the prediction was made. This is the "time to prediction".
3. If the event is not found, record a "sector miss". Also record the difference between the time the event was predicted to occur and the time at which the prediction was made. This is the "time to prediction".
4. Delete from the data any aircraft whose id is specified in the "reject_acids" file.
5. Group the individual prediction error/time to prediction values into a format for plotting. For this analysis, we grouped pairs into 5 minute buckets based on time to prediction; all pairs which describe a time to prediction of between 0 and 4 minutes are grouped into the first bucket, 5 to 9 in the next bucket, and so on.
6. For each bucket, set the sector hit rate to the ratio of the number of "sector hits" in the bucket to the number of "sector hits" plus "sector misses".
7. On the x axis of the plot is "time to prediction" for each bucket, on the y axis is "hit rate" (as a percent) for each bucket, or the result as described in the previous sentence. This is the plot of sector hit rate.

Notes: Steps 1 – 5 are handled by an analysis program "event compare". Steps 6 – 8 are handled by simple "Octave" scripts. Both tools are discussed in the appendices.

This process results in a plot of sector hit rate. It, together with sector count prediction, indicate the overall accuracy with which a tool predicts sector counts on the aggregate and at the level of individual flights.



This plot shows the sector hit rate for TFAS and ETMS, proposed and active flights for the August 11 data sample. For active flights, hit rates start near 70 and 80 percent for ETMS and TFAS, respectively, and gradually ramp down. For proposed ETMS flights, hit rate appears to stay between about 45 and 75 percent, for TFAS proposed flights hit rate appears to stay between about 45 and 75 percent, though on average over the time period TFAS performs better than ETMS. TFAS appears to perform better than ETMS by about 10 percent for active flights.

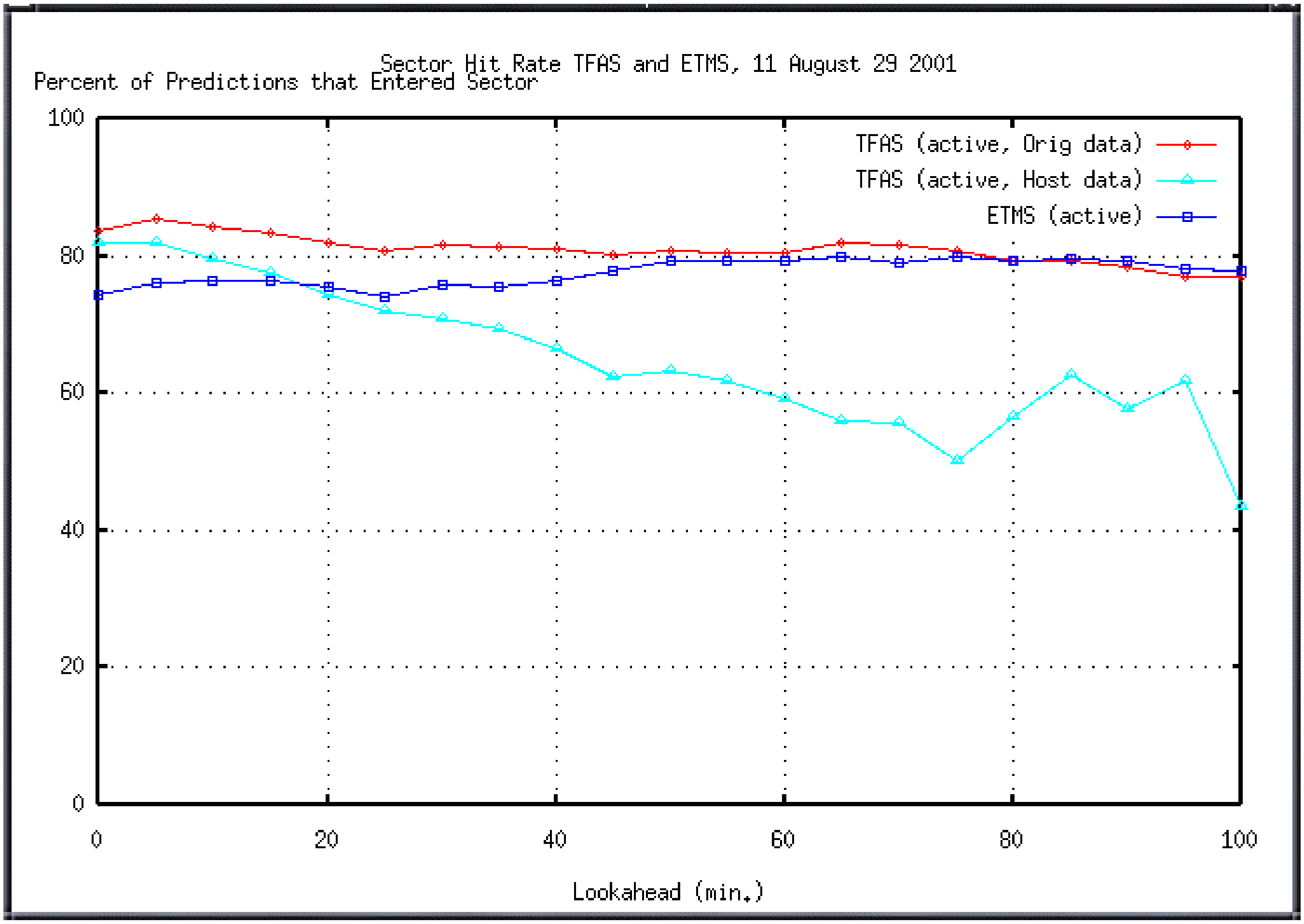
Data Sample Characteristics
ETMS

Number actual ac 2237 (acs which make appearance during time period)
Number actual ac after filtering 2190
Number predicted ac 2453 (acs which make appearance during time period)
Number predicted ac after filtering 2148
Number actual events 11440 (valid ac only)
Number predicted events 40449 (valid ac only)

TFAS

Number actual ac 2237 (acs which make appearance during time period)
Number actual ac after filtering 2190
Number predicted ac 2380 (acs which make appearance during time period)
Number predicted ac after filtering 2075

Number actual events 11440 (valid ac only)
Number predicted events 191568 (valid ac only)



This plot shows the sector hit rate for TFAS and ETMS, TFAS using both Orig and Host data, for the August 29 data sample. For longer lookahead times, ETMS and TFAS appear to perform similarly. The most notable thing about this plot is that TFAS using Host data appears to perform much worse than either of the other cases. Part of this can be attributed to the small number of flights at longer lookaheads, but the low prediction quality for short lookaheads deserves further investigation.

We would expect hit rate to be higher for shorter lookaheads and fall off for longer. The severe weather on the 29th may account for the "flat" characteristics of these plots.

Data Sample Characteristics:
ETMS:

Number actual ac 4193 (acs which make appearance during time period)
Number actual ac after filtering 3898
Number predicted ac 4640 (acs which make appearance during time period)
Number predicted ac after filtering 3554
Number actual events 20040 (valid ac only)

  - Number predicted events 92113 (valid ac only)

- TFAS (Orig):
  - Number actual ac 4193 (acs which make appearance during time period)
  - Number actual ac after filtering 3898
  - Number predicted ac 4525 (acs which make appearance during time period)
  - Number predicted ac after filtering 3439
  - Number actual events 20040 (valid ac only)
  - Number predicted events 325856 (valid ac only)

Projections of TFAS Performance Under Certain Operational Circumstances

One of the main strengths of the TFAS programs is that it was developed from the NASA Traffic Management Advisor (TMA) baseline. All of the descent optimizing and meter fix scheduling algorithms have been incorporated into the TFAS software baseline.

Were TMA is run operationally, the amount of delay each aircraft needs to absorb before crossing the meter fix in order for the flow of aircraft to be at a desirable level can be shown on the controllers scopes. Hence, controllers will intercede with the nominal flow of aircraft that need to be delayed.

The effect of the use of TMA scheduling information on TFAS sector accuracy will be investigated by comparing the accuracy of the sector entry times into the TRACON by arrival aircraft using the unmodified estimated time in the TRACON against the estimated time in the TRACON plus 80% of the amount of delay the aircraft needed to absorb.

The amount of delay each aircraft needed to absorb comes from the archived operational TMA cm_sim file. This information was provided by NTX CTAS personnel for each day of our analyses. In the operational TMA cm_sim file, the Estimated Time of Arrival (ETA) an Scheduled Time of Arrival (STA) to the meter fix is written out in the RECORD_DATA line. The format of the line follows:

```
RECORD_DATA 73 301 TM_MSG TIM AAL1424/AMA.0908    FK10 BAMBE 999083729
999086486 999086500 999086500 999085892 999085906 999085906 0 0 5 13 999086402 0 5
999086500
```

The third filed represents the relative time of the estimates. The sixth field indicates the aircraft identification. While the 11th and 12th fields hold the ETA and STA for the aircraft respectively. The amount of delay needed to be absorbed is max(STA-ETA-2,0). The TRACON is capable of absorbing approximately 2 minutes of delay for any aircraft, so this is taken out of the delay needed to be absorbed by the last ARTCC sector.