

AATT Research Project

Distributed Air Ground (DAG) CE-7 En Route Traffic Flow Management (TFM) Control Degree of Freedom Analysis

Reference Contract: NAS2-98004
from NRA2-36648(BRM)
in response to:
Request No. AATT RTO # 60

Prepared for:
NASA/Ames Research Center
Acquisition Division
Attn.: Nellie M. Powell
Contracting Officer
JAZ-241-1
Moffett Field, CA 94035-1000
November 1, 2000

Work performed by
James A. Rome
Ronald W. Lee
Simon D. Rose

Oak Ridge National Laboratory, Oak Ridge, TN 37831

Contract Managed by
James Cistone

Lockheed Martin Air Traffic Management
Rockville, MD 20850
October 2001

Table of Contents

1.	Introduction and motivation.....	1.1
2.	Data sources and caveats.....	2.1
2.1.	ETMS Data Processing.....	2.2
3.	Characterization of the airspace in the large.....	3.1
3.1.	Cylinder plots.....	3.1
3.2.	Flow plots.....	3.3
3.3.	Sector busyness.....	3.3
4.	Characterization of airspace in the small: ZOB48.....	4.1
4.1.	Cleveland Center—ZOB48.....	4.1
4.2.	Interactions with adjacent sectors.....	4.3
4.3.	Flows by origin and destination.....	4.7
4.4.	Complexity metrics.....	4.9
4.5.	Cause of ZOB48 complexity.....	4.16
5.	New approaches to sector traffic management.....	5.1
5.1.	4-D deconfliction.....	5.1
5.2.	Assumptions.....	5.7
5.3.	Simulation.....	5.8
5.4.	Restricted airspace.....	5.10
5.5.	Extra planes.....	5.14
5.6.	Application of these techniques to ATM.....	5.17
6.	Conclusions and recommendations.....	6.1
	Appendix A. Data Cleaning problems on 1 August 2001.....	A-1
	References.....	R-1

Purpose of the work

The goals of the NASA Advanced Air Transportation Technologies (AATT) Research Task Order (RTO) 60 were to enable both analysis of current conditions and development of innovative strategies for rerouting and spacing of air traffic to alleviate the congestion for local Traffic Flow Management (TFM). This control degree of freedom analysis will form a key component of the Distributed Air/Ground (DAG) Traffic Flow Management concept¹. DAG-TM has a goal for beyond 2015 for gate-to-gate operations to enhance user flexibility and efficiency, and to increase system capacity, without adversely affecting safety or restricting accessibility in the National Airspace System (NAS). DAG-TM will be implemented by development of fourteen concept elements (CEs) that cover all operational domains of all phases of flight. DAG CE-7 involves en route departure, cruise and arrival phases and will require collaboration for mitigating local Traffic Flow Management (TFM) constraints due to weather, Special Use Airspace (SUA) and complexity².

Current problems arise from inefficient use of en route airspace in the presence of bad weather, SUA and complexity. Deviations in the congested airspace are frequently excessive and not preferred by the users. Solutions to this problem may involve system-wide collaboration between the Air Traffic Service Provider (ATSP) and users at the Flight Deck (FD) level as well as the Aeronautical Operational Controls. The three stages of the solution include preemptive user action, collaboration on local TFM initiatives and TFM actions.

The work performed provides the initial concept for strategies that would fit all the three stages (preemptive user action, collaboration on local TFM initiatives and TFM actions) of the solutions to congestion. The team has used a practical approach that has leveraged ATM work performed for the FAA and NASA. RTO 60 consists of six principal tasks:

- Task 1 - Understanding the Background
- Task 2 - Define Current Traffic Situations
- Task 3 - Determine Strategies
- Task 4 - Analyze Strategies
- Task 5 - Evaluation and Conclusion
- Task 6 - Expand Assessment

The two primary objectives of the work were to develop:

- A high-fidelity estimation of both the delay impact on flights and inefficiencies in current-day practices and procedures;
- A methodology to evaluate the benefit mechanisms related to the proposed operational methods for applying rerouting and spacing in innovative ways to alleviate local congestion effectively and efficiently;

Secondary objectives were originally set to develop:

- A qualitative assessment of the extension of more optimal contributions of rerouting and spacing to the national Air Traffic Control System Command Center (ATCSCC) strategies level; and
- A qualitative assessment of the dynamic SUA degree of freedom for local and national (ATCSCC) TFM applications including recommendations for further study.

While performing this task, discussion with NASA Ames staff Wendy Holforty, Steven Green, and Becky Grus during a team Teleconference on July 12, 2001 resulted in a consensus that it would be beneficial to redirect the focus of the work for RTO 60. The task was refocused on the development of strategies for rerouting and spacing to improve congestion that involve analysis of a single sector. It was agreed that the busy and complicated air traffic congestion in the airspace of the Cleveland Air Route Traffic Control Center (ARTCC) would be appropriate. Consequently, Sector ZOB48 was selected as one of the most complicated and busy sectors in the NAS.

The work involved initial use of the Post Operation Evaluation Tool (POET) to broadly characterize ZOB48. Team members also met and discussed the congestion problems and TFM needs with Mr. Dan Wiita of the Cleveland ARTCC, who controls ZOB48. We felt that the present complexity of ZOB48 is at the limit of what controllers can manage using present techniques and tools. An indication of this limit is that when the sector is busy, requests from the Traffic Management Unit (TMU) and filed routes are usually ignored.

The method we developed uses real flight data and an improved four-dimensional (4-D) deconfliction display to simulate traffic management in the sector. The “as-flown” Enhanced Traffic Management System (ETMS) traffic can be managed in the sector, and most flights can get through the sector on a shorter path.

We used our methods to maintain the sector traffic load even when a significant portion of the sector is closed, simulating a bad weather cell or a closure of SUA.

We were successfully able to change the way a sector was managed by creating and reserving private interference-free “pipes” for each aircraft. Once the usual sector traffic has been accommodated, we were able to add extra pipes through the sector at many times in the day, representing additional flights between popular origin-destination pairs.

The techniques developed by the team could be employed at the ARTCC console, in the plane cockpits, and in a more global planning system. Proposed solutions for congestion may therefore be effective at the three stages of TFM: preemptive user action, collaboration on local TFM initiatives and TFM actions. The techniques developed by the team could be incorporated as part of the Constrained Airspace Tool (CAT) being proposed as a solution to the problem².

En Route Traffic Analysis and Control at the Sector/Center Level

1. Introduction and motivation

Ever since the time of Icarus, man has dreamed of being able to fly like a bird. Man learned to fly, but (aside from hang gliders and ultralights) we do not fly like a bird. This is worth considering because huge flocks of birds fly and wheel in the air. The birds all stay together, they do not hit each other, and there are no air traffic controllers to guide them. The rules that govern flocks of birds are very simple, and there is a huge literature³ of simulated bird flocks (called “boids”). Boids only look at their nearest neighbors and try to match velocity with these neighbors while not hitting them. But air traffic at the sector level is totally unlike a flock of birds. At any given time, only a few planes have the same origin or destination. As a result, it is not sufficient to consider just nearby planes; the total ensemble of flights have to be considered all the time in order to get every one safely to their destinations.

In 1990–91 we performed *An Analysis of the National Airspace Capacity* for the FAA and Congress⁴. As part of the study, we calculated the density in the airspace around Chicago and found that the busiest part of the space was at cruising altitudes West of Pittsburgh. This is exactly the area that has proved to be a bottleneck during the past several summers. Although in that report of nine years ago, we found that the bottlenecks all occurred at the airports, today it appears that some restrictions to flow of aircraft are also imposed by the airspace itself.

Flow constraints would be expected to occur when parts of the airspace are closed, usually due to a weather event or to SUA restrictions. These restrictions might appear as either an increase or a decrease in the usual traffic because planes diverted from a closed region may increase the density elsewhere. Detecting airspace restrictions from position fixes (such as the TZ messages of the ETMS) poses a challenge. On the other hand, restrictions might also arise from the inability of controllers to handle the volume and complexity of traffic in a sector.

In this study we attempt to characterize the local nature of air traffic and then also study what is probably the most complicated and busiest sector in the world, ZOB48 in the Cleveland Center. We will attempt to characterize the traffic in this sector to see why it is complicated. Then we will try to find ways to enhance and improve the traffic flow in this sector.

2. Data sources and caveats

There are some major caveats that must be applied to this work, and they also help to explain some of our approaches. We had to rely upon historical ETMS data that is readily obtainable from the FAA. We originally hoped to use several weeks of data from the summers of 1998 and 1999. These data worked well for characterizing overall sector busyness, but because it only had position messages (TZs) at four-minute intervals, the resolution was insufficient to use for detailed sector analysis. Therefore, we obtained data from the summer of 2001 (July 29 –August 3) that has one- and two-minute TZ messages.

The data obtained from the FAA is imported into an Oracle database where it can then be queried. In general, the interval between TZ messages is greater than one minute. Therefore, we interpolate the results of our database queries to obtain one-minute TZs. We also must filter these data (not always successfully) to replace the altitudes that are the “fly to” goal of a flight, and not its current altitude. For example, a flight taking off from Cleveland (CLE) might have the altitudes in its initial TZ messages as 210 (in hundreds of feet), which is clearly impossible. Sometimes these TZs are labeled with an altitude type of “T,” but often, the altitude type is not specified. We attempt to replace these bogus altitudes with interpolations from nearby “good” altitudes. In addition, we have found that sometimes the speeds are also incorrect.

The resolution of TZ data is also an issue. Times are specified to the nearest minute, and positions (latitude and longitude) also to the nearest minute. We note that a plane going at 360 knots goes 6 nm per minute, and that a minute of longitude is one nautical mile (nm) on the equator.

As an example of the problem, if we replay the cleaned TZ data for a day on ZOB48, we typically get over 30 “separation violations,” which are defined as closer than 5 nm spacing and 1000 ft (2000 ft above 29000 ft) altitude separation. If we interpolate between the minute data points, we get even more violations. Clearly this is a data artifact and is incorrect; we try to keep this in mind in our analysis. *Appendix A* lists the incorrect perceived separation violations in our cleaned ETMS data on 1 August 2001.

The ETMS data tables are separated by message type and by day the flight leg originated. Therefore, in order to know all flights in the air at a given time and place, queries must be combined from the day of interest and the end of the previous day.

We also have no knowledge of whether a flight is being controlled by a given sector, even if it is physically within the sector’s three-dimensional (3-D) airspace. Handoffs may occur up to 20 miles on either side of a sector boundary. There are also classes of traffic that are customarily handled by adjacent sectors (and that never get handed off), even when they are physically within another sector. In our previous capacity study, we had access to the Aircraft Management Program (AMP) data sets, which contained the actual handoff times. If these data were available, the handoff time could be used to se-

lect the correct set of TZs. Lacking the handoff times, we were forced to use the sector boundary crossings.

Our challenge is to try to use poor-resolution, possibly incorrect, and not necessarily the best available data to infer performance of the air space and to attempt to improve it.

2.1. ETMS data processing

ETMS data were received from the FAA as Oracle dump files and loaded into an Oracle instance running locally. Data for two time periods were received, 30 July 1999–19 August 1999 and 28 July 2001–03 August 2001. Direct database queries soon were discovered to be insufficient for the analysis processing necessary for this study for two reasons: the queries (especially writing out the answers) were very slow, and the data required post-processing.

The ETMS data cleaning exercise began with a query to create views that joined the *tz_data* table records with *flight_data* to obtain the origin and destination airports by Greenwich Mean Time (GMT) day.

```
CREATE OR REPLACE VIEW TZACT_&&date AS
  SELECT
    t.acid,
    f.dept_aprt,
    f.arr_aprt,
    f.act_date,
    t.flight_index,
    t.POSIT_TIME,
    t.CENTER,
    t.CUR_LAT,
    t.CUR_LON,
    t.GROUNDSPEED,
    t.ALTITUDE,
    t.altitude_type
  FROM tz_data_&&date t, flight_data_&&date f
  WHERE
    t.acid=f.acid
    AND t.flight_index = f.flight_index
    AND t.act_date = f.act_date
    AND t.cur_lat BETWEEN 1500 AND 3000
    AND t.cur_lon BETWEEN 3900 AND 7500
  ORDER BY t.ACID, t.posit_time;
```

Step 1: Dump and extract raw data

ETMS stores data in tables for each GMT day, a flight leg's departure time determining its day. Since there are overlaps across day boundaries a query for a single day's data must merge tables for the target and previous days. Therefore, although we had 7 days of ETMS data, we could only get complete data sets for the last 6 days. The SQL query we used to query our views was:

```
select
  acid,
  dept_aprt,
  arr_aprt,
  flight_index,
```

```

        posit_time,
        center,
        cur_lat,
        cur_lon,
        groundspeed,
        altitude,
        altitude_type,
        act_date
from tzact_&&date
union
select
        acid,
        dept_aprt,
        arr_aprt,
        flight_index,
        posit_time,
        center,
        cur_lat,
        cur_lon,
groundspeed,
        altitude,
        altitude_type,
        act_date
from tzact_&&yesterday
where
        posit_time >= to_date('&&yesterday 20','yyyymmdd hh24')
        and
        altitude > 0
order by acid, posit_time;

```

Query result records were saved in a comma-delimited ASCII format, and sorted by aircraft id, flight index, and time stamp, the former two fields comprising a flight key. An example record is below. Fields in order are: aircraft id, departure airport, arrival airport, flight index, time stamp year, month, day, hour, minute, center, latitude in minutes north, longitude in minutes west, ground speed, altitude, altitude type.

```

AAL1,JFK,LAX,166484,2001,08,03,13,53,W,2376,4604,481,350,

```

The resulting files are quite large, especially the data from 2001 which included more frequent TZ messages. Given our focus on ZOB48, we extracted from each day file records with positions within a bounding box around Cleveland Center, 86W,38N to 75W,45N. Sizes of files compressed with the GNU Zip utility and record counts of the original and extracted data are shown in the Table 2.1.

Table 2.1. Sizes for 2001 Data.

Date	Raw		Cleveland Center	
	Compressed Size (bytes)	Record Count	Compressed Size (bytes)	Record Count
29 Jul 2001	46,013,663	4,105,964	6,612,413	567,672
30 Jul 2001	50,765,359	4,595,014	7,973,445	701,052
31 Jul 2001	52,726,891	4,759,352	8,177,232	714,294
01 Aug 2001	53,966,307	4,872,894	8,308,953	725,023
02 Aug 2001	52,221,305	4,867,975	7,823,053	709,706
03 Aug 2001	52,856,849	4,927,306	7,869,807	713,641
Total	308,550,374	28,128,505	46,764,903	4,131,388

Step 2: Fix raw data

Two problems with ETMS data were addressed in this step. First, position reports (TZ records) are needed each minute for each active flight to support sector-level analysis. TZ messages in 1999 data occur every four minutes, and although 2001 data included one-minute TZs for some flights, many had gaps of two minutes or more. Second, altitude values are often erroneous.

With the data sorted by flight and time, we were able to process a flight by loading all its records. To fix the altitudes, we first detected and marked bad altitudes, and then attempted to improve them. Three tests for bad altitudes were applied. First, we check for an initial “climb to” altitude by comparing first and second altitudes for a flight. If the second is more than 5000 feet below the first, we assume a “climb to” and replace the first altitude and ground speed values with those from the second record. Second, any altitude with type *T* is marked as bad. Third, we run a series of altitude value comparisons for flights with more than two records. For each record pair compared, if the earlier altitude type is not *C* (*C* altitudes are known to be good), and the delta between the altitudes is greater than the threshold of 5000 feet, we assume one of the two altitudes is bad and must then determine which of the two to mark as bad.

In describing the comparisons, we represent the current or later record as *t*, the earlier or previous as *t-1*, the next record as *t+1*, the record before the previous as *t-2*, and so on. If the altitude type for *t* is *C*, *t-1* is assumed bad. If *t* is the last record for the flight, altitude deltas are computed and compared for *t-2*, *t-1* and *t-1*, *t*. Else, if *t* is not the first record for the flight, deltas for *t-2*, *t* and *t-1*, *t+1* are compared. Otherwise, we check if *t*'s altitude is greater than both *t-1* and *t+1*. If so, we compare *t-1*, *t+1* to *t-1*, *t*. If not, we fall through to a comparison of *t-1*, *t* and *t*, *t+1*. If the first of each pair of values compared is less than the second, the current (*t*) altitude is marked as bad, otherwise the previous (*t-1*) is bad.

Fixing bad altitudes is a matter of finding the nearest good altitudes occurring before and after a bad altitude. If prior and subsequent good values exist, we interpolate between them linearly based on the record time stamps. If the record to be fixed is not between

records with good altitudes, it is assigned from the closest earlier or later value not marked bad, whichever exists. If all records for a flight have bad altitude values, no changes to the values are made. Each modified altitude value is marked with a lower case *i* appended to the altitude type field.

Finally, if time stamps for successive records for a flight span more than one minute, records are created at each minute with linearly interpolated ground speed, latitude, longitude, and altitude values. Interpolated records are indicated with an altitude type value of an upper case *I*.

An example of altitude value repair is given below for flight AAL1002 on 03 August 2001. A snippet of original records from the raw data follows:

```
AAL1002,ORD,BWI,156643,2001,08,03,12,26,W,2373,4770,465,298,C
AAL1002,ORD,BWI,156643,2001,08,03,12,26,W,2373,4770,465,298,C
AAL1002,ORD,BWI,156643,2001,08,03,12,27,W,2372,4760,465,271,C
AAL1002,ORD,BWI,156643,2001,08,03,12,28,W,2370,4751,447,250,
AAL1002,ORD,BWI,156643,2001,08,03,12,29,W,2369,4742,432,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,30,W,2369,4732,432,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,31,W,2368,4724,426,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,32,W,2367,4715,412,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,33,W,2366,4707,408,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,34,W,2365,4699,388,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,35,W,2365,4691,375,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,36,W,2365,4683,355,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,37,W,2366,4676,355,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,38,W,2366,4668,355,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,39,W,2366,4660,351,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,40,W,2366,4653,345,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,41,W,2366,4646,345,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,42,W,2367,4638,354,150,T
AAL1002,ORD,BWI,156643,2001,08,03,12,44,W,2369,4622,289,96,
AAL1002,ORD,BWI,156643,2001,08,03,12,48,W,2367,4602,289,86,
```

Note the last two fields of each record, the altitude and altitude type, respectively. AAL1002 has several good altitude values before and after a string of type-‘T’ records. Clearly, the flight didn’t drop from 25000 feet to 15000 feet in a single minute, fly at 15000 for 14 minutes and then drop to 9600 feet. After repair, the following records are produced:

```
AAL1002,ORD,BWI,156643,996841560,W,2373,4770,465,298,C,
AAL1002,ORD,BWI,156643,996841620,W,2372,4760,465,271,C,
AAL1002,ORD,BWI,156643,996841680,W,2370,4751,447,250,-,
AAL1002,ORD,BWI,156643,996841740,W,2369,4742,432,240,Ti,
AAL1002,ORD,BWI,156643,996841800,W,2369,4732,432,230,Ti,
AAL1002,ORD,BWI,156643,996841860,W,2368,4724,426,220,Ti,
AAL1002,ORD,BWI,156643,996841920,W,2367,4715,412,210,Ti,
AAL1002,ORD,BWI,156643,996841980,W,2366,4707,408,201,Ti,
AAL1002,ORD,BWI,156643,996842040,W,2365,4699,388,191,Ti,
AAL1002,ORD,BWI,156643,996842100,W,2365,4691,375,182,Ti,
AAL1002,ORD,BWI,156643,996842160,W,2365,4683,355,172,Ti,
AAL1002,ORD,BWI,156643,996842220,W,2366,4676,355,163,Ti,
AAL1002,ORD,BWI,156643,996842280,W,2366,4668,355,153,Ti,
AAL1002,ORD,BWI,156643,996842340,W,2366,4660,351,144,Ti,
AAL1002,ORD,BWI,156643,996842400,W,2366,4653,345,134,Ti,
AAL1002,ORD,BWI,156643,996842460,W,2366,4646,345,125,Ti,
```

```

AAL1002,ORD,BWI,156643,996842520,W,2367,4638,354,115,Ti,
AAL1002,ORD,BWI,156643,996842580,W,2368,4630,322,106,I,
AAL1002,ORD,BWI,156643,996842640,W,2369,4622,289,96,-,
AAL1002,ORD,BWI,156643,996842700,W,2369,4617,289,94,I,
AAL1002,ORD,BWI,156643,996842760,W,2368,4612,289,91,I,
AAL1002,ORD,BWI,156643,996842820,W,2368,4607,289,89,I,
AAL1002,ORD,BWI,156643,996842880,W,2367,4602,289,86,-,

```

The 14 minutes at 15000 feet are replaced with a linear descent from 25000 to 9600. Further, we see interpolated one-minute records inserted where missing between the final three records of the original data. In this case, we believe the repaired data to be a more accurate representation of the flight's path and the cleaned data are much more useful for sector-level analysis. Of course, there are flights in the original data for which erroneous altitudes were not detected in the checks described above and some which could not be repaired (e.g., all altitude types are *T*).

Step 3: Match to sector/FPA/module boundaries

Sector analysis is only possible if flight messages and data can be associated with the sector that is controlling the flight. This information is not recorded in ETMS data, so it must be inferred. We accomplished this by matching positions against sector geographic boundaries. Sectors are arranged in a hierarchy with one or more fix posting areas (FPAs) in a sector and one or more modules in an FPA. Of particular interest are the points, times, and positions at which flights cross from one sector/FPA/module to another. Output from Step 2 was passed through another filter process for matching records to sector boundaries and computing transitions between sectors. Although we can track transitions between FPAs and modules, sectors suffice, especially since many of the sectors of interest (e.g., ZOB48, ZOB27, ZOB49, ZOB66) consist of a single FPA and module. Below is a snippet of records for DAL1003 from Portland, ME to Cincinnati on 03 Aug 2001.

```

DAL1033,PWM,CVG,148864,996837660,C,2479,4850,451,310,I,
DAL1033,PWM,CVG,148864,996837720,C,2474,4858,451,310,-,
DAL1033,PWM,CVG,148864,996837780,C,2469,4866,454,310,I,
DAL1033,PWM,CVG,148864,996837840,C,2464,4873,456,310,-,
DAL1033,PWM,CVG,148864,996837900,I,2459,4881,456,310,-,
DAL1033,PWM,CVG,148864,996837960,C,2454,4888,462,310,-,
DAL1033,PWM,CVG,148864,996838020,I,2450,4895,460,310,-,
DAL1033,PWM,CVG,148864,996838080,C,2446,4905,438,310,-,
DAL1033,PWM,CVG,148864,996838140,I,2445,4914,453,310,-,
DAL1033,PWM,CVG,148864,996838200,I,2443,4924,459,310,-,
DAL1033,PWM,CVG,148864,996838260,I,2436,4930,435,310,-,
DAL1033,PWM,CVG,148864,996838320,I,2430,4937,453,310,-,
DAL1033,PWM,CVG,148864,996838380,I,2425,4944,459,310,-,
DAL1033,PWM,CVG,148864,996838440,I,2419,4951,459,299,Ti,
DAL1033,PWM,CVG,148864,996838500,I,2414,4959,466,288,Ti,

```

After sector matching, these records produce the following output. Line continuations are indented, and sector transitions are in boldface.

```

DAL1033,PWM,CVG,148864,996837660,C,2479,4850,451,310,I,-,?,ZOB5700M1,?,?,?,
-1,230.2607,0,0,-,
DAL1033,PWM,CVG,148864,996837720,C,2474,4858,451,310,-,-,?,ZOB5700M1,?,?,?,

```

```

-1,230.2967,0,0,-,
DAL1033,PWM,CVG,148864,996837780,C,2469,4866,454,310,I,ZOB5700M1,273.23,
ZOB4800M1,69.53,41.1893,-81.0371,310,230.3325,0,996837751,-,
DAL1033,PWM,CVG,148864,996837840,C,2464,4873,456,310,-,-,?,ZOB4800M1,?,?,?,
-1,226.5675,0,0,-,
DAL1033,PWM,CVG,148864,996837900,I,2459,4881,456,310,-,-,?,ZOB4800M1,?,?,?,
-1,230.4038,0,0,-,
DAL1033,PWM,CVG,148864,996837960,C,2454,4888,462,310,-,-,?,ZOB4800M1,?,?,?,
-1,226.6398,0,0,-,
DAL1033,PWM,CVG,148864,996838020,I,2450,4895,460,310,-,-,?,ZOB4800M1,?,?,?,
-1,232.9626,0,0,-,
DAL1033,PWM,CVG,148864,996838080,C,2446,4905,438,310,-,-,?,ZOB4800M1,?,?,?,
-1,242.2023,0,0,-,
DAL1033,PWM,CVG,148864,996838140,I,2445,4914,453,310,-,-,?,ZOB4800M1,?,?,?,
-1,261.7039,0,0,-,
DAL1033,PWM,CVG,148864,996838200,I,2443,4924,459,310,-,-,?,ZOB4800M1,?,?,?,
-1,255.2691,0,0,-,
DAL1033,PWM,CVG,148864,996838260,I,2436,4930,435,310,-,-,?,ZOB4800M1,?,?,?,
-1,213.0659,0,0,-,
DAL1033,PWM,CVG,148864,996838320,I,2430,4937,453,310,-ZOB4800M1,213.04,
ZID8701M1,33.67,40.5862,-82.1827,310,221.5942,0,996838268,-,
DAL1033,PWM,CVG,148864,996838380,I,2425,4944,459,310,-,-,?,ZID8701M1,?,?,?,
-1,226.8469,0,0,-,
DAL1033,PWM,CVG,148864,996838440,I,2419,4951,459,299,Ti,-,?,ZID8701M1,?,?,?,
-1,221.6717,0,0,-,
DAL1033,PWM,CVG,148864,996838500,I,2414,4959,466,288,Ti,-,?,ZID8701M1,?,?,?,
-1,230.7192,0,0,-,

```

Fields are the same as Step 2 output through altitude type. Additional fields in order are:

- sector exited
- exit azimuth (azimuth from crossing point to the center of the exit sector in degrees)
- entry sector
- entry azimuth
- crossing point latitude (linearly interpolated)
- crossing point longitude (linearly interpolated)
- crossing point altitude (linearly interpolated)
- heading (great circle heading computed from successive records)
- cross vertical flag (-1 means descent, 1 ascent, 0 horizontal transition)
- crossing time (linearly interpolated)

In the records above, we see DAL1033 was heading southwest when it crossed from ZOB57 to ZOB48 air space at level flight. The point of boundary crossing occurred at 41.1893N, 81.0371W, with an altitude 31000 feet, a heading of 230.3325 degrees, and at time 996837751 (in seconds past 1 January 1970), which is 31 seconds after the last position report in ZOB57 and 29 seconds before the first position report in ZOB48.

After some time in ZOB48, DAL1033 crosses into ZID87 air space. These crossing points, interpolated and inferred from ETMS data, were used to determine the periods of time during which flights were controlled by individual sectors. We know this is not exact, because handoffs occur as far as 20 miles from a sector boundary, and some flights through a sector's air space are not controlled by the sector at all. However, this is a reasonable approximation using ETMS data. Given the size of a typical sector [e.g., ZOB48 is roughly 140 miles wide (East–West) and 90 miles (North–South)] and the time a flight

spends in a sector (about eight minutes in ZOB48), the inferred crossing points are useful for analyzing sector traffic.

In summary, we were required to fix the altitudes in the TZ messages in order to be able to determine sector crossings. The cleaning process described above, is imperfect, but better than doing nothing. In Section 5, we use the sector crossings to attempt to simulate a realistic traffic flow. Some of these flights will be incorrect, but this does not materially affect our conclusions.

3. Characterization of the airspace in the large

Traffic flow in the airspace is inherently a 4-D problem in space and time. There are several underlying questions that one would like to answer:

- Is a region of airspace crowded? Compared to what?
- Is a sector crowded? Compared to what?
- Is there a correlation between weather and busyness?
- What are the flows in the space?

We approached these questions in several ways.

3.1. Cylinder plots

Cylinder plots are especially useful for studying the spatial flows centered on an airport. As shown in Fig. 3.1, traffic descends from the en-route altitudes to the major airports (only one of close-by airports is labeled to avoid interference). However, the more interesting area is the airspace above Cleveland where the density is up to 5.0 plane-minutes/100 square miles. We will examine this region in more detail in Section 4.

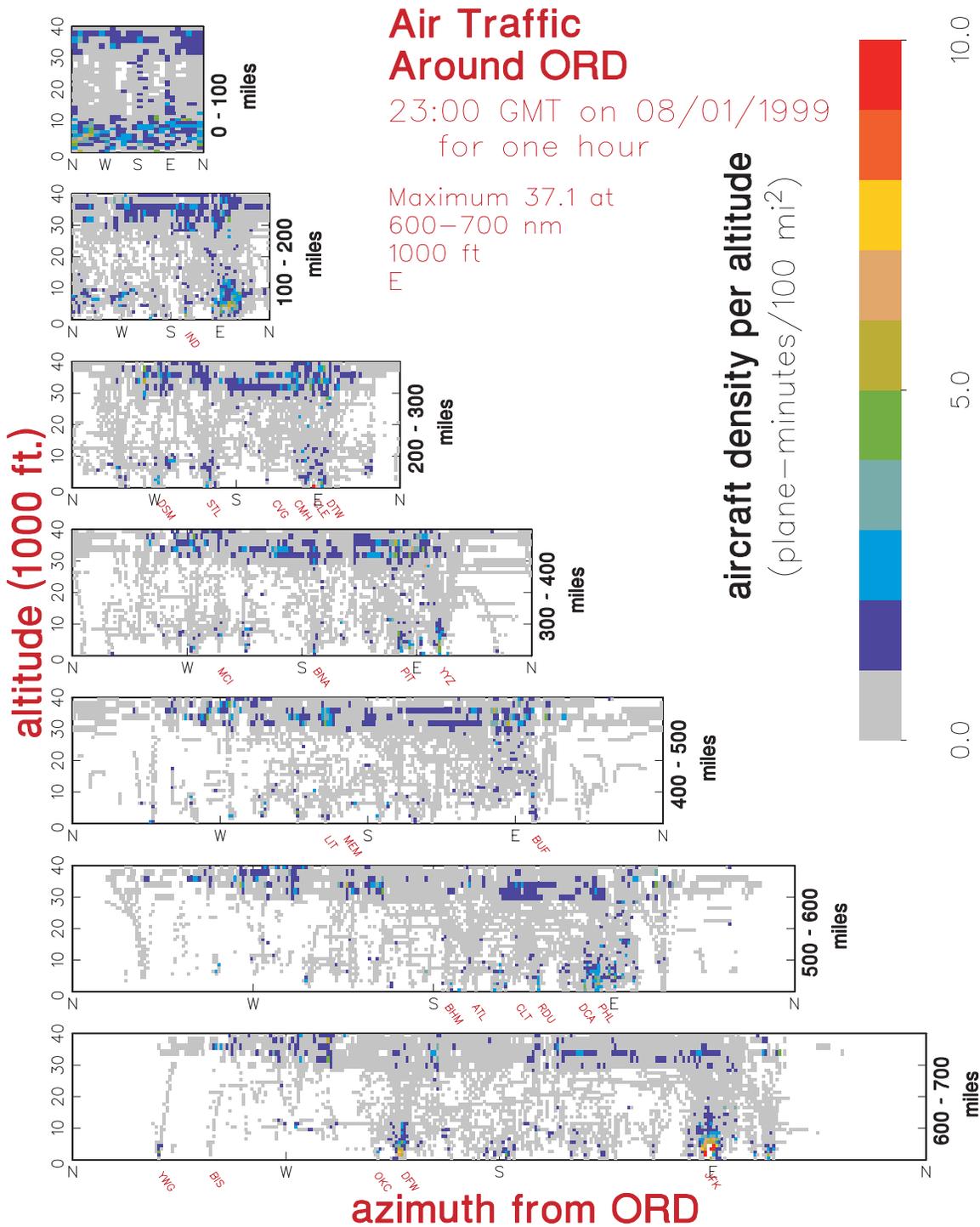


Fig. 3.1. Traffic density in concentric cylinders around Chicago. Each cell in the unwrapped cylinders has the same number of square miles. White areas have no planes in them. Some major airport locations are shown in red. At this time, the heaviest concentration of traffic is near the New York City Airports. The colors represent the number of plane-minutes in each hundred square miles.

3.2. Flow plots

It is not just the air space density that is important, but also the fluxes. We attempted to display the densities and the flows as is shown in Fig. 3.2. The plane density is highest in New York City and Chicago. During this hour, it appears that there is a net influx of traffic to the New York region, presumably due to scheduling and bank effects. In many places, there are large flows in both directions, so we felt that the resolution in the plot was inadequate to obtain a quantitative feel for what is happening.

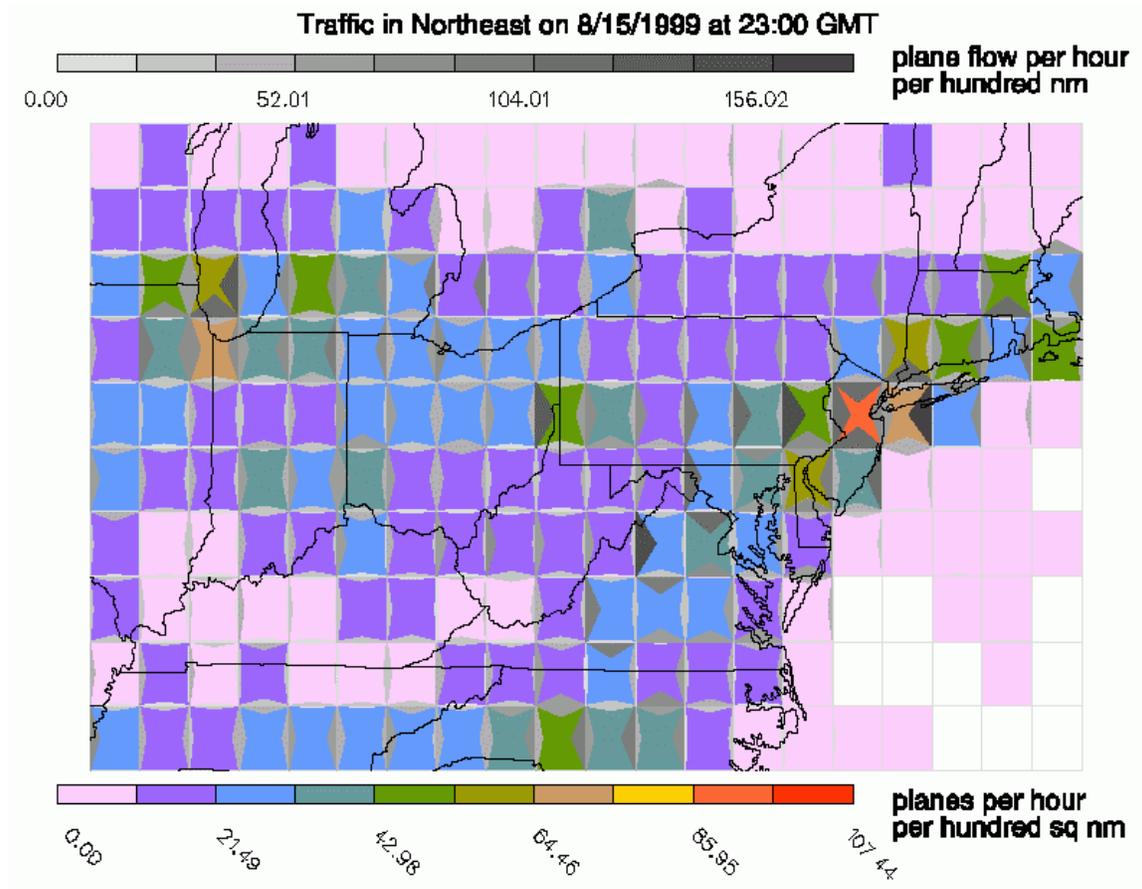


Fig. 3.2. Air traffic densities and flows in the Northeast. The inward-pointing triangles represent the flows into each side of the cells. The height of the triangle and color are proportional to the flux (top legend). The density in the cell is keyed to the bottom legend.

3.3. Sector busyness

Instead, we decided that it was essential to focus on the sector for several reasons:

- They are the fundamental FAA air traffic management entities.
- They have wildly varying geometry, size, and function that do not correspond with predefined areas or volumes such as were used in Fig. 3.1 and Fig. 3.2.

But we had a strong desire to see both the overall flow of traffic in the airspace, while at the same time being able to drill down and see the traffic details in a given sector. To this end we developed a new representation, a detail of which is shown in Fig. 3.3.

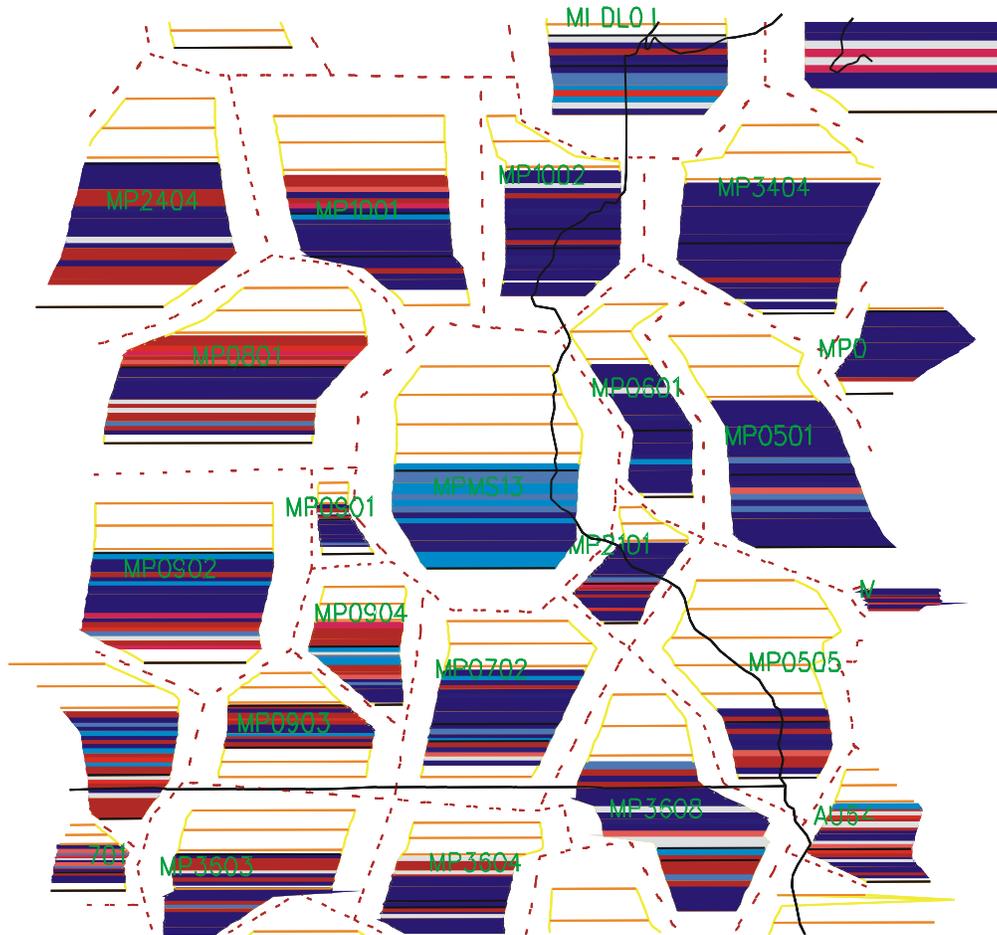


Fig. 3.3. Detail of sector traffic plot. The plot displays the sector Fix Posting Area (FPA) outlines at a given altitude (here 10000 ft and near Minneapolis) as red dashed lines. Within each of these is a slightly smaller plot (outlined in yellow) that represents the sector plane density at each altitude. The horizontal orange lines occur every 5000 ft. Because these are low sectors, the traffic is only plotted up to the sector top. The density is indicated at each altitude level (1000 ft up to 29000 ft, and 2000 ft above that).

Fig. 3.4 shows a larger view of the Northeast traffic density. New York City and Chicago are the busiest, and there is also significant traffic density around the major airports at Saint Louis (STL), Detroit (DTW), Washington, and Charlotte (CLT). The same data are plotted in Fig. 3.5, but here the color corresponds to the deviation from a 21-day average. There is a clear correspondence to the weather pattern at the time as is shown in Fig. 3.6. The traffic has been diverted around the weather system. We can understand Fig. 3.5 by examining how individual flights were affected by the bad weather on July 31, 1999. To do this, we compare the same flight on two days (good and bad) and connect the flight's positions at the same time relative to the start of the day. This technique allows us to not only see how the flight was deviated, but also its lateness. The result is shown in Fig. 3.7.

Sector Traffic in Midwest at Altitude 10000 ft

Max hourly density/10 nm²=101.4

on 19990731 at 0 GMT

Modules at this altitude: 879
Total sectors: 1031

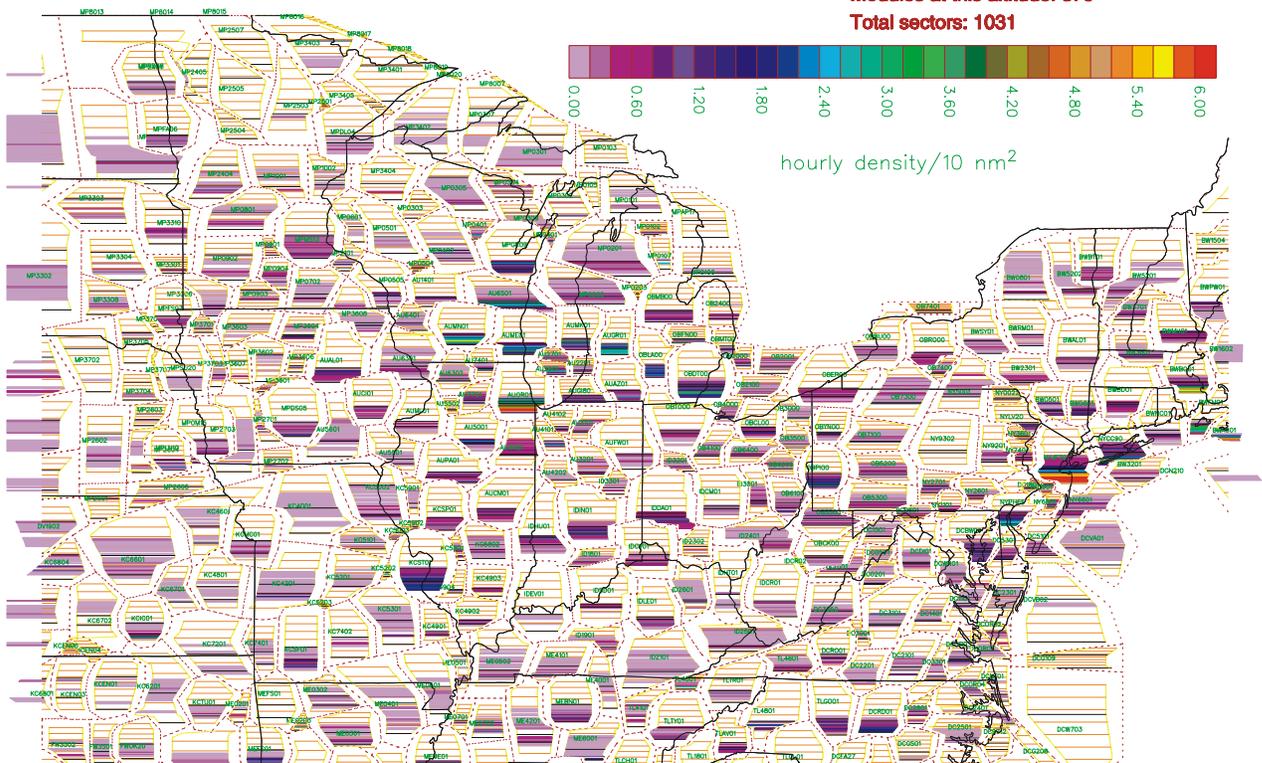


Fig. 3.4. Traffic density at low altitudes at 0:00 GMT of 31 July 1999. The New York City area is busiest, followed by Chicago.

Sector Traffic in Midwest at Altitude 10000 ft
on 19990731 at 0 GMT

Modules at this altitude: 879
Total sectors: 1031

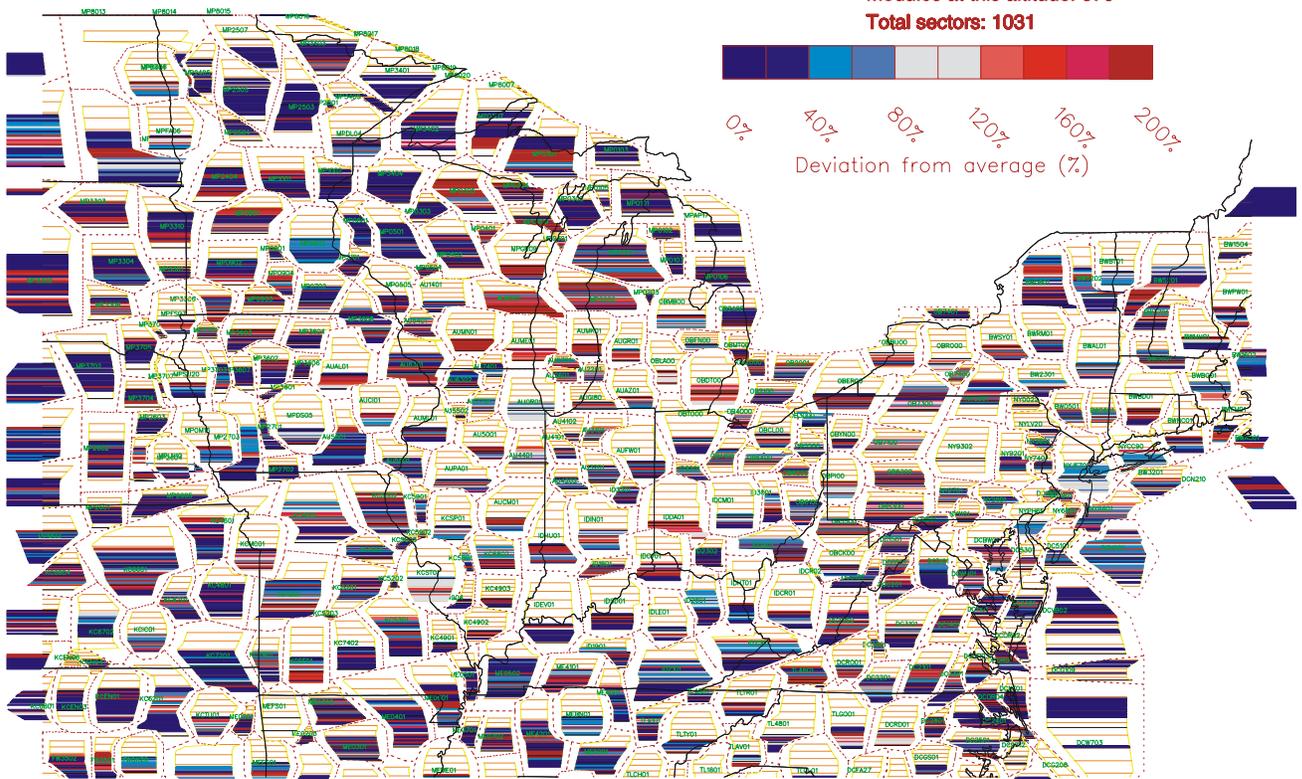


Fig. 3.5. The same plot as in Fig. 3.4 except that the colors correspond to the deviation from a 21-day average. The area around MSP and Western Wisconsin is much lower than average, while the area north of Chicago is much heavier than average.

Sector traffic at 10000 feet
0:00 GMT on 7/31/1999

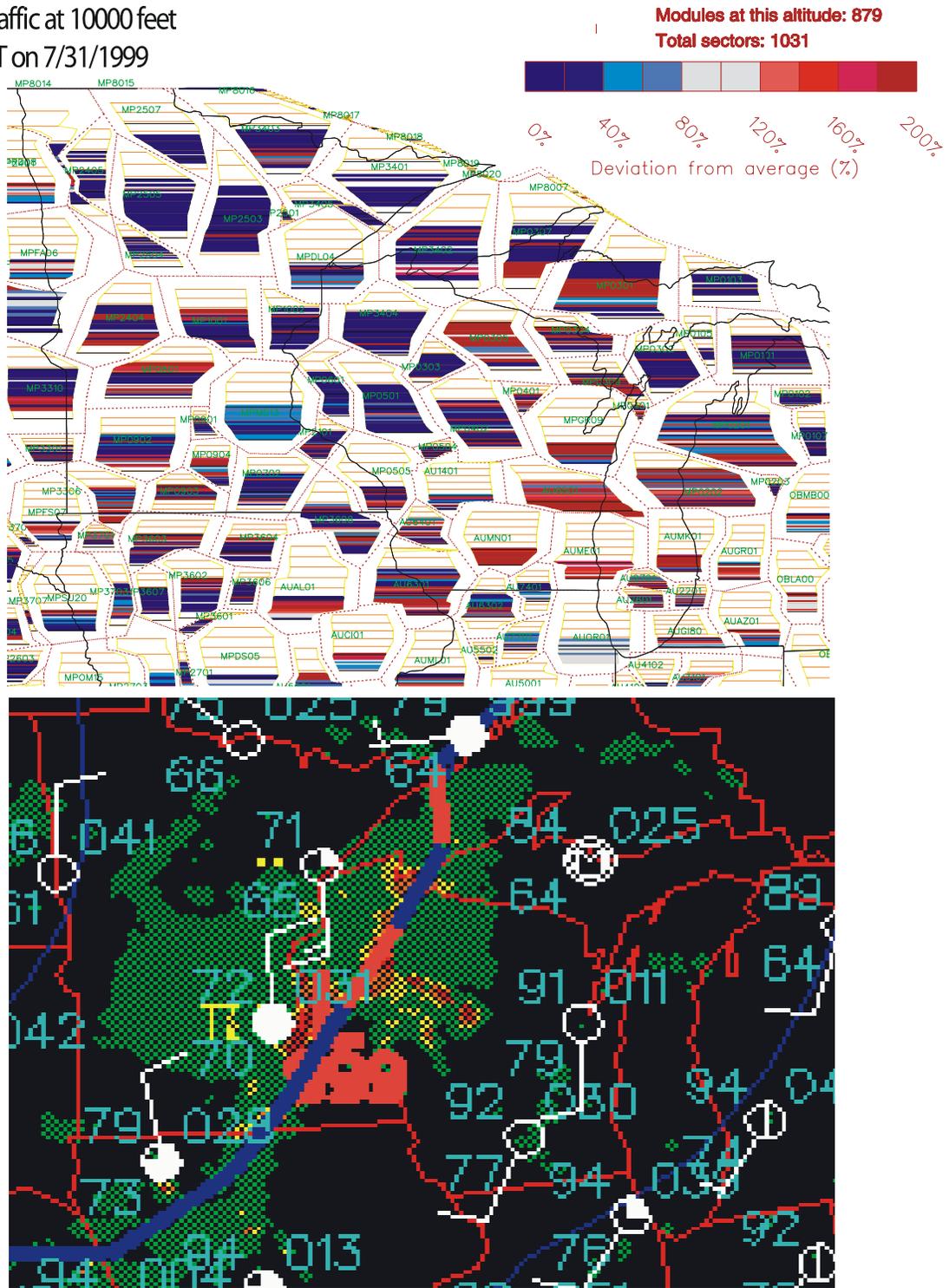


Fig. 3.6 The relative plane density (top) shows a clear correspondence to the weather pattern at the time (bottom). The density is much lower than normal where the weather is bad, and higher than normal in adjacent sectors to which traffic has been diverted.

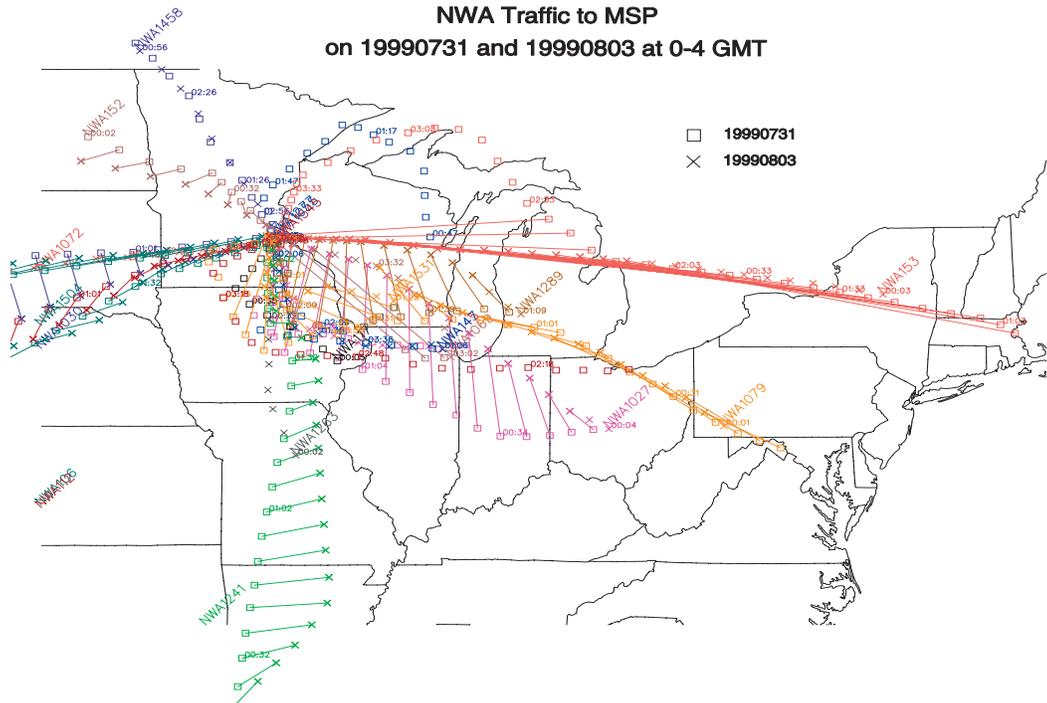
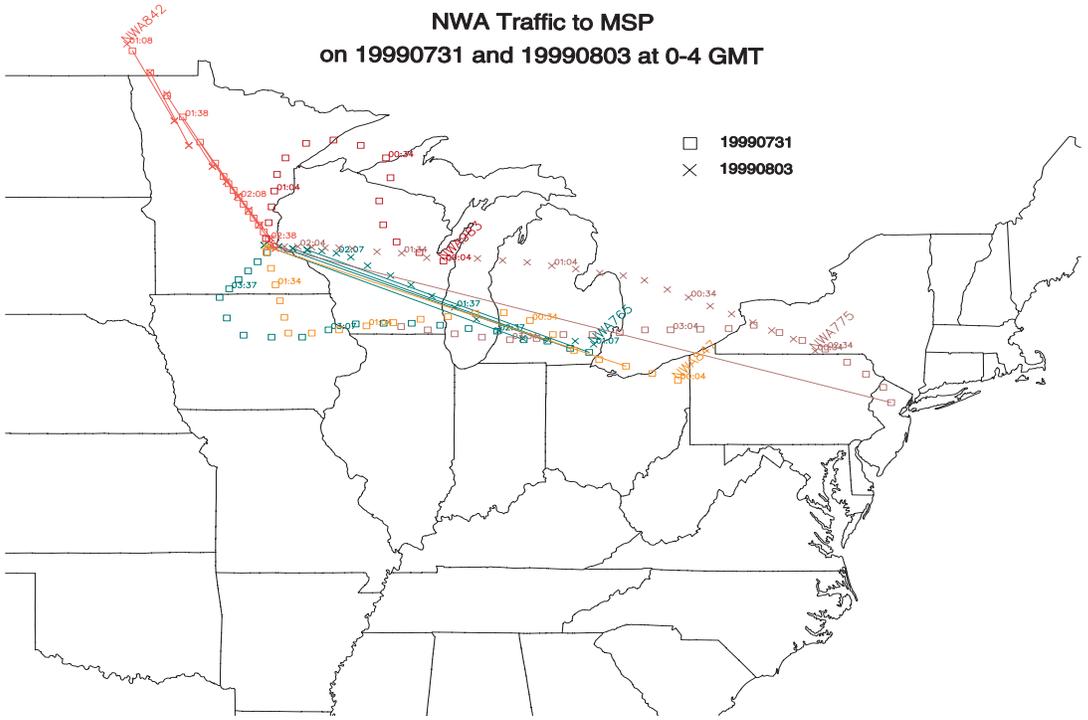


Fig. 3.7. Comparison of the same flight flow on a bad day (7/31) and a good day (8/3). To avoid the bad weather, flights were vectored North or South of the disturbance. Fights coming in from the West were relatively unaffected.

4. Characterization of airspace in the small: ZOB48

We decided to focus on one sector, and picked ZOB48 — a difficult one. There have been many studies relating operational errors to sector characteristics, and an excellent review of this is given in the paper by Rodgers, Mogford, and Mogford⁵. We are interested in how complexity limits the throughput of a sector, rather than in operational errors; nonetheless all of the factors cited in that review can come into play in ZOB48 (referred to by the Center as the Ravenna sector).

4.1. Cleveland Center—ZOB48

Air traffic controllers in the Cleveland ARTCC have the highest pay scale of any center due to the volume, density, and complexity of air traffic through ZOB. Arguably the busiest and most difficult to control sector in the world is ZOB48, the Ravenna sector of Cleveland Center, for which controllers must have 15–20 years of experience. ZOB48 is a high sector handling traffic between flight levels 240 and 329 (from 24,000 to 33,000 feet). The traffic through ZOB48 has increased about 30% between 1999 and 2001.

Although density and flow volume contribute to controller workload, it's the complexity of ZOB48 that makes its control difficult. Figure 4.1 shows the routes through Cleveland center.

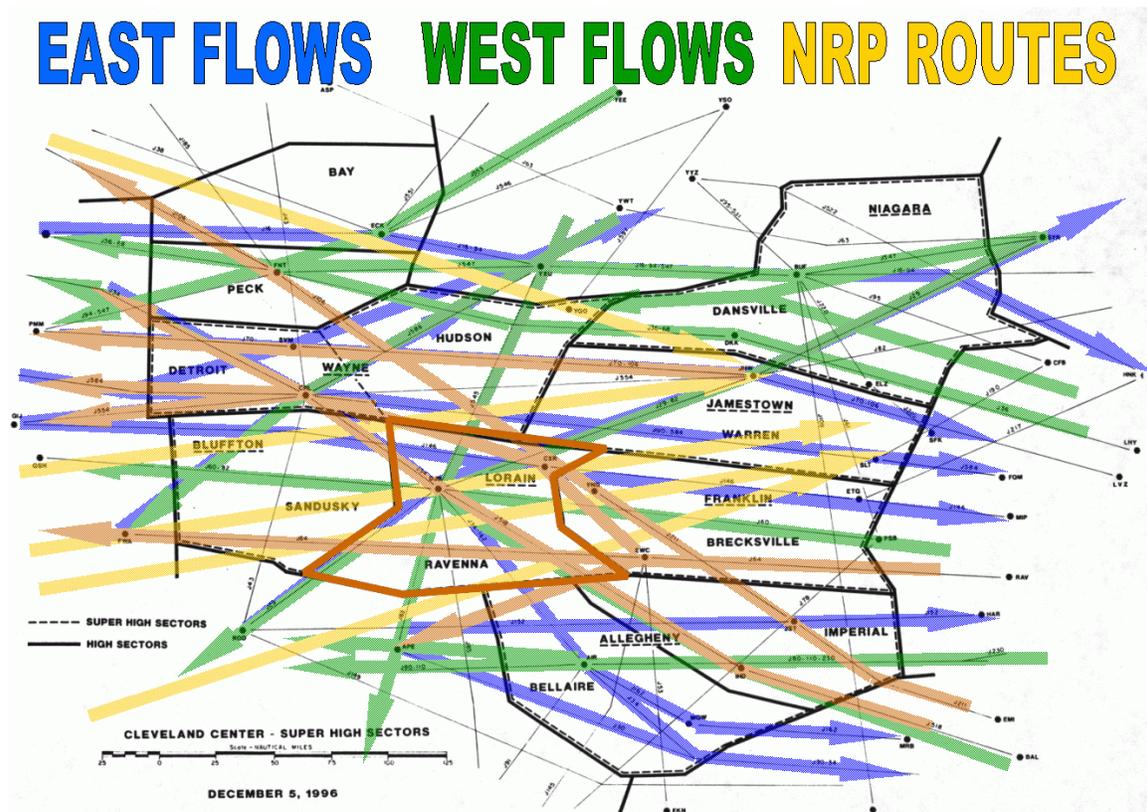


Fig. 4.1. Routes through Cleveland Center. ZOB48 and ZOB49 are the high and super high sectors labeled Ravenna and Lorain, respectively and are outlined in red. (Image courtesy of the Cleveland ARTCC.)

Congestion in ZOB48 is principally a matter of geography — its location along the routes to many major airports. Visible in Fig. 4.1 are the many crossing routes for major traffic flows:

- Arrivals and departures for Detroit (DTW).
- Arrivals and departures for Pittsburgh (PIT).
- Departures for Cleveland (CLE)
- Arrivals and departures for Cincinnati (CVG).
- Organization of eastbound flows to the New York City airports and westbound flows to Chicago.
- Southern traffic to and from Toronto.
- Traffic between the Midwest and Philadelphia (PHL) and the three Washington DC airports.
- Other over flights.

The sector route geometry is made clearer in Fig. 4.2, which shows the routes taken by planes leaving CLE. The pink lines are the jetways, many of which intersect at a point just to the west of Cleveland.

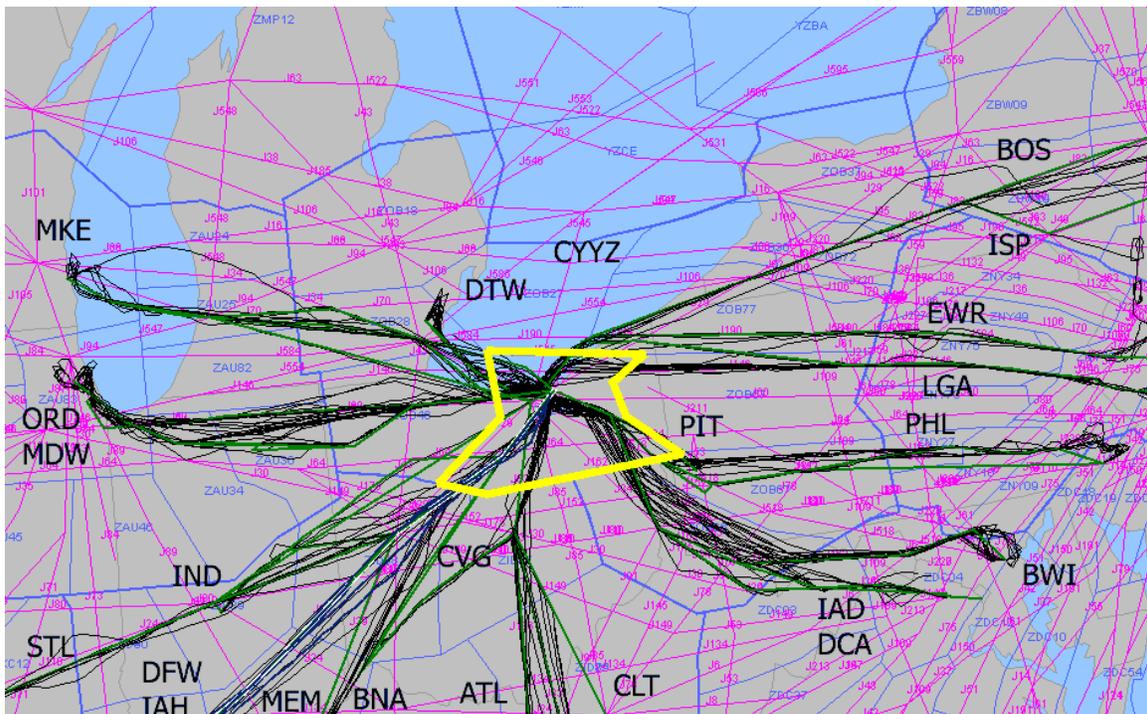


Fig. 4.2. Flows, merge points, and dispersion points for traffic through ZOB48 (yellow). The major jetways are shown in pink. There is a major jetway merge point in the middle of ZOB48 (at the DRYER Navaid), at the center of these tracks. (Display generated using the Post Operations Evaluation Tool, POET, developed by Phil Smith of Ohio State University and Metron staff).

Many flows cross at the DRYER Navaid (DJB at 82.16W, 41.36N) just to the northwest of the middle of the sector. The ZOB48 situation is made worse because traffic is generally kept South of Canadian airspace, although the Cleveland center does control a portion of Canadian airspace over Lake Erie. What cannot be seen in this figure is that a significant portion of the traffic through ZOB48 is climbing or descending.

Altitude transitions are the most difficult task for controllers to manage while deconflicting airspace, and contribute heavily to complexity measures. Ascents and descents occur at typically at 1000–2000 feet per minute. ZOB48 controllers maintain a 4-D picture of all the traffic in their heads and plan for the handling of aircraft as soon as they receive a handoff, which may be as many as 20 miles from either side of the sector boundary. All aircraft must be deconflicted long before they reach crossing points such as DRYER⁶.

However, the controller's mind view starts to fall apart when he/she is distracted. Example sources of distraction include:

- A pilot does not understand a controller command, and it must be repeated until it is acknowledged.
- A pilot asks a question at a busy time.
- An adjacent sector controller does not accept a handoff or is not prompt about it. In this case, the controller cannot allow the plane to exit the sector, and there is essentially no space for circling in ZOB48.
- An airline files a flight to cut across one of the major "standard" traffic flows, e.g., the PIT departure stream. This requires more attention, diverting the controller's attention from other flights and effectively reducing the sector throughput.

ZOB48 is physically adjacent to more than 20 other sectors (the principal ones are shown in Fig. 4.3), and controllers spend as much as 40% of their time communicating with controllers of other sectors, usually by phone. When things are really busy, we observed a third controller at the ZOB48 station who handled communications. This dramatically improves sector operation. However, the physical arrangement of the sector control station forced the third controller to stand and to reach over the heads of the other two controllers to reach the necessary controls.

4.2. Interactions with adjacent sectors

Flows between the air space of sectors adjacent to ZOB48 as derived from ETMS data for the period of 29 July 2001 to 03 August 2001 are depicted in Fig. 4.4 –Fig. 4.6. Table 4-1 describes the adjacent sectors, and the sector boundaries are illustrated in Fig. 4.3.

Table 4-1. Flows between ZOB48 and its neighbors

Sector	Location Relative to ZOB48	Flows Relative to ZOB48
ZID87	High, South	Entries > Exits
ZID88	High, Southwest	Entries = Exits
ZOB27	High, North	Entries > Exits
ZOB35	Low, East	Entries << Exits
ZOB40	Low, Northwest	Entries << Exits
ZOB46	High, West	Entries > Exits
ZOB49	Super High, Above	Entries = Exits
ZOB57	High, East	Entries > Exits
ZOB64	Low, Below	Entries > Exits
ZOB66	High, Southeast	Entries << Exits
ZOB67	High, Southeast	Entries = Exits
ZOB77	High, Northeast	Entries < Exits

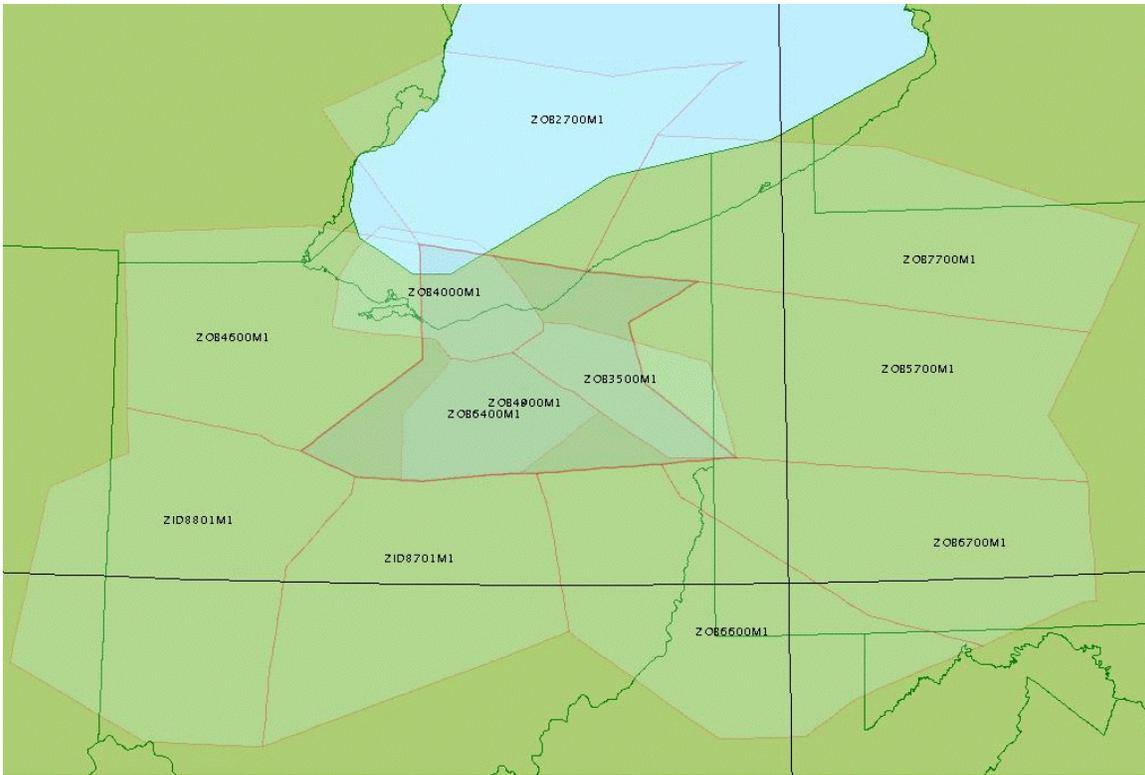


Fig. 4.3. Sectors interacting with ZOB48.

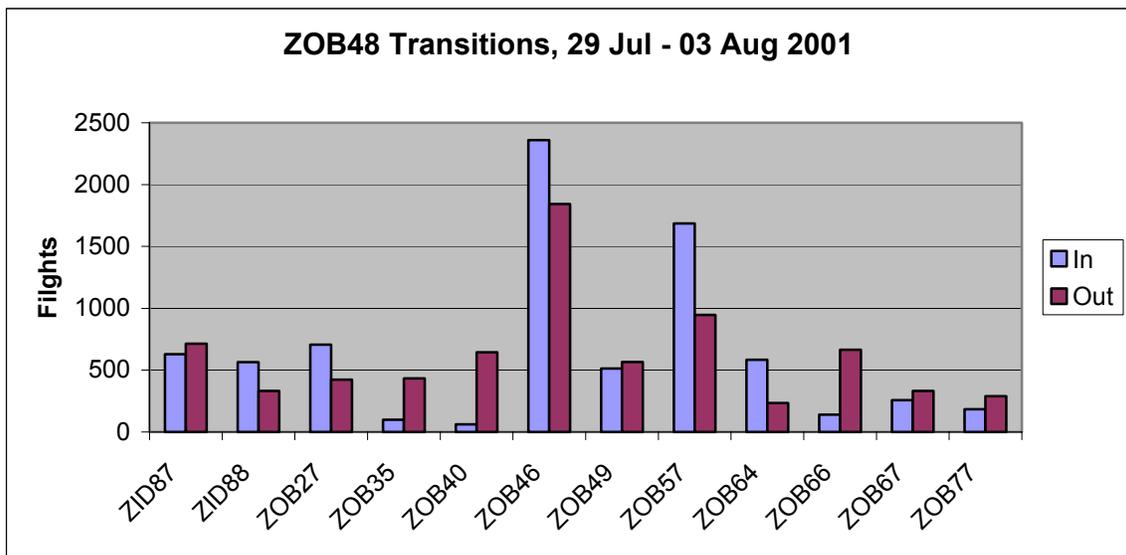


Fig. 4.4. Transitions in and out of ZOB48 air space by sector, 29 July 2001 – 03 August 2001. Most traffic is exchanged with ZOB46 and ZOB57.

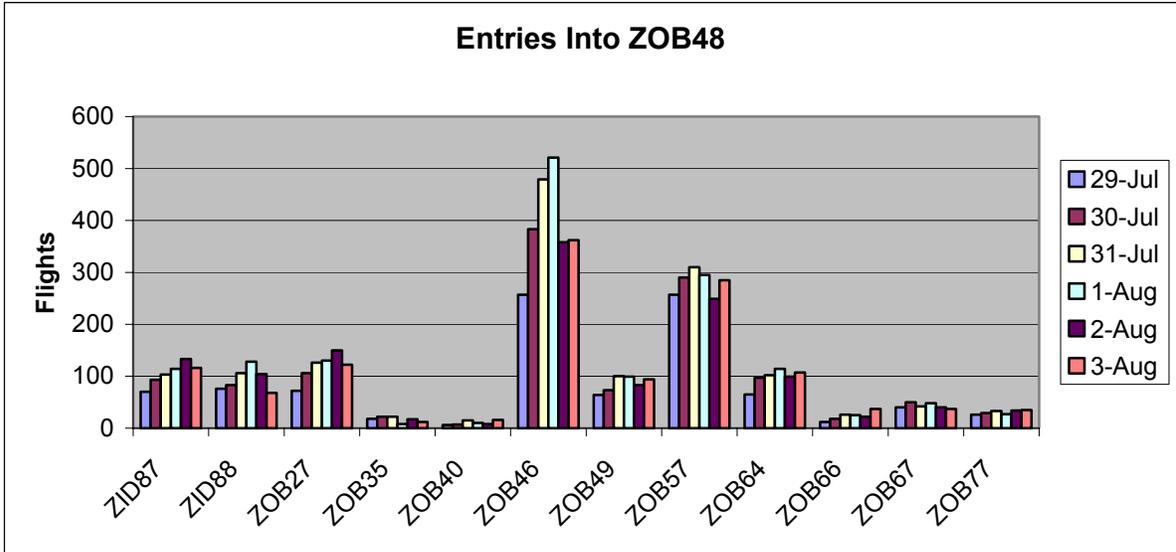


Fig. 4.5. Daily flights entering ZOB48 air space by sector, 29 July 2001 – 03 August 2001. August 1 was a Wednesday.

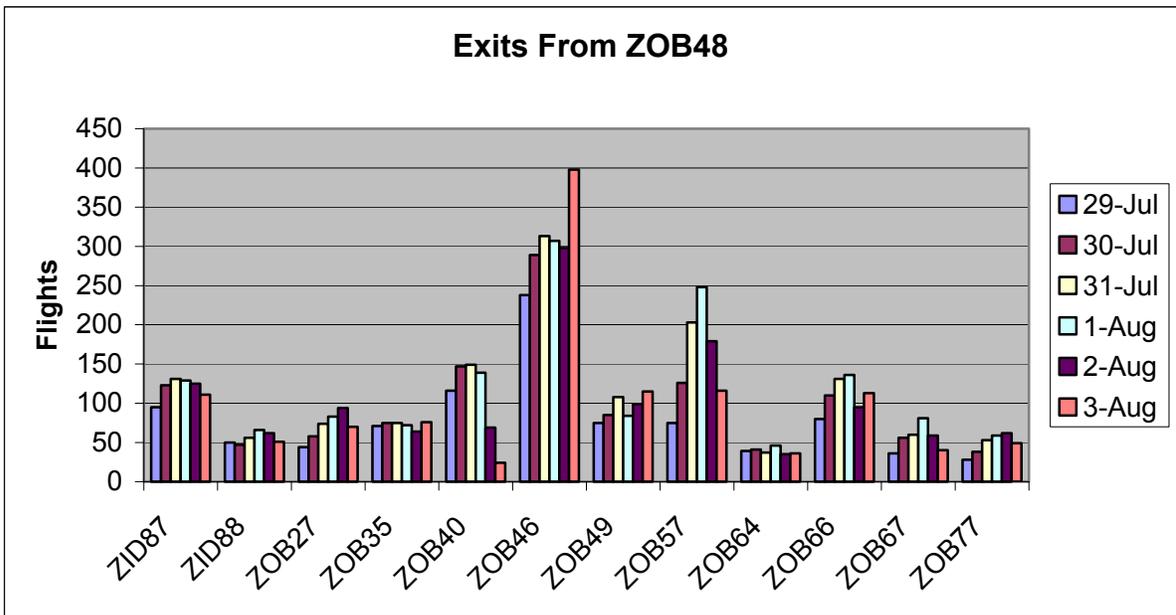


Fig. 4.6. Daily flights exiting ZOB48 air space by sector, 29 July 2001 – 03 August 2001.

Note the transitions represented in the above figures are based on published geographical boundaries of the modules comprising the fix posting areas of the identified sectors and are not recorded handoffs between sector controllers. Handoff information is not available in ETMS and therefore must be inferred. In addition, the quality problems of ETMS data, especially dubious altitude values, add uncertainty when assigning a TZ position to a sector. We preprocessed the ETMS data to interpolate to one-minute TZs when necessary and try to recognize and fix errant altitude values as was described in Section 2.1, *ETMS Data Processing*.

Nonetheless, boundary crossings are reflective of traffic flows in the general sense in spite of specific exceptions. There are flights that occupy ZOB48 air space but are not managed by ZOB48 controllers and vice versa⁶. For example, many Cleveland airport (CLE) departures are handled by lower sectors even when flying in ZOB48 altitude levels, and Detroit (DTW) departures cutting the southwest corner of ZOB48 are not handed off to ZOB48. Clearly, ZOB46 and ZOB57 provide the lion's share of hand-offs to and from ZOB48.

4.3. Flows by origin and destination

ZOB48's proximity to several major airports and location along the Midwest-to-Northeast corridor are the source of its many conflicting routes. A breakdown of flows by origin and destination highlights the complexity of the sector. The data shown below are cumulative for the six days from 29 July 2001 to 03 August 2001.

Departures

Leading departure airports contributing flights crossing into ZOB48 airspace are shown below. Departures from CLE are not included since many of them are not handled by ZOB48.

Origin	Flights	Feed	Flights	Exit	Flights
DTW	1288	ZOB27	183	ZOB66	188
		ZOB30	119	ZID87	172
		ZOB46	90	ZOB57	139
		ZOB41	84	ZOB67	74
		ZOB21	77	ZOB49	55
		ZOB35	61		

Entry mostly from the northwest and north, some ascending from lower sectors. Exit mostly to points southwest, southeast, and east with some ascents into ZOB49 above.

PHL	959	ZOB57	459	ZOB46	284
				ZOB40	61
				ZOB27	45

Entry from the east. Exit mostly with some northwest and descents.

CVG	831	ZID88	273	ZOB27	99
		ZOB46	77	ZOB77	90
				ZOB46	51
				ZOB57	56
				ZOB32	53
				ZOB49	52

Entry mostly southwest. Exits west and north with some northeast.

PIT	684	ZOB64	185	ZOB46	296
		ZOB66	54		
		ZOB62	53		

Entry from the southeast. Exits west.

CYYZ	433	ZOB27	193	ZID87	94
				ZOB49	35
				ZOB46	33

Entry from the north. Exit mostly south with some west and ascents.

Dispersion points for departure streams occur various places. PIT traffic enters at 81.07W,40.65N and fans out at 81.56W,40.78N. Traffic bound for ORD fans out at 84.44W,41.82N, and IND-bound traffic disperses at 84.05W,40.23N. Two streams bounds for DTW enter around 81.77W,41.74N and 83.17W,40.90N, respectively.

Arrivals

Leading arrival airports contributing flights crossing into ZOB48 airspace are shown below. Note that no CLE arrivals pass through ZOB48

Destination	Flights	Feed	Flights	Exit	Flights
DTW	1614	ZID87	395	ZOB40	473
		ZOB57	392	ZOB46	225
				ZOB64	71

Entry from the south and east. Exit mostly descending northwest, some due west.

PIT	928	ZOB46	422	ZOB35	344
				ZOB40	72

Entry from the west. Exit descending southeast and early descents in northwest portion of ZOB48 air space.

LGA	918	ZOB46	188	ZOB57	413
		ZOB27	78	ZOB49	35
		ZID88	75		
		ZOB30	35		

Entry mostly west with some north and southeast and some ascents from below. Exit mostly west with ascents into ZOB49.

ORD	853	ZOB57	216	ZOB46	425
		ZOB64	66		

Entry mostly from the east with some ascents from below. Exit due west.

CVG	487	ZOB27	97	ZID87	216
		ZOB77	66		

Entry from the north and northeast. Exit south.

EWR	477	ZID88	96	ZOB57	107
		ZOB46	33	ZOB77	61
		ZID87	27	ZOB49	37
		ZOB64	25		

Entry mostly southwest with some west and south and some ascents from below. Exit mostly west with some northwest and some ascents into ZOB49.

CYYZ	448	ZOB46	82	ZOB27	135
------	-----	-------	----	-------	-----

ZID88	82	ZOB46	53
ZOB67	23	ZOB57	31

Entry mostly west and south with some from the southeast. Exits are mostly north with some northwest.

There are several merge points for arrival traffic. DTW arrivals merge near 82.20W,41.38N . LGA merges occur at 81.20W,41.53N and 78.30W,41.24N, the latter for traffic from the south. EWR arrival streams merge at 78.20W,41.56N with southeast traffic passing through DRYER. PIT arrivals merge at 81.30W,41.14N, proceed to 80.64W,40.90N, and exit around 80.44W,40.69N.

4.4. Complexity metrics

Common measures of air space complexity include aircraft density and flux versus time. Neither captures some significant aspects of controller workload, specifically altitude transitions and aircraft spatial proximity. Metrics accounting for these critical factors were developed and applied against ETMS data for the period of 29 July 2001 to 03 August 2001.

Aircraft activity

A more useful measure of activity in a region of air space, as opposed to density or volume, is the amount of effort a controller must expend in order to manage the associated traffic, i.e., the controller workload. Although controller workload is not captured implicitly in ETMS or other flight-oriented data, perhaps it can be inferred. Reference 5 discusses many of the controller workload factors as they apply to controller errors. We have attempted to look at them explicitly, and also using the 4-D deconfliction methods explained in Section 1.

As shown in Fig. 4.7, there are many ways to view controller activities. We believe that controllers must do something when the following events occur:

- Handoffs to and from the sector
- Changes in course (vectoring)
- Changes in altitude
- Changes in speed
- Collision avoidance

Of course, without explicit knowledge of the controller's actions, we must determine when one of these changes occurs using ETMS data (with its already discussed limitations). To do this, we fitted the original data (colored) with at most two straight lines. The junction of these lines determined the time (and place) of a controller's action. However, the track of the plane (shown at the top right) cannot always be fitted by just two straight lines. Square boxes in the figure mark these more complicated fits. The colors for each track are selected by doing a hash of the origin and destination to obtain a color, so each origin-destination pair should have the same color.

The bottom two plots show the speed and altitude of each flight along with the (gray) fits to the ETMS data. Note the large percentage of flights that are climbing or descending in ZOB48. The junctions of the fits and the sector entrance/exit times are determined and plotted in stacked bars for each minute in the activity plot. Along the top of the plot are orange lines representing the number of close encounters (less than 1.5 times the usual collision avoidance distance and altitude separation) between planes during each minute.

Finally, the black curve on the activity plot is derived from the 4-D deconfliction display derived in Section 5. The display of Fig. 5.5 is calculated at each minute for each active flight. If there are other planes within $\pm 45^\circ$ of the plane's current heading, and within the sector's altitude range, the fraction of colored space is added to the complexity for that minute. This metric seems to be higher during the collision avoidance events. It seems low during the activity spike at 23:00. We believe that the proximity of other flights along the future path of a given flight should represent an enhanced workload for the controller that is not measured in the other individual activity metrics.

22:00 to 24:00

ZOB48 on 20010801

Origin= *, Destination= *

complexity

Handoffs = 304

Altitude Changes = 88

Speed Changes = 116

Vectoring Events = 141

5-min Average Activity (+10)

Collision avoidances

Number in sample = 165

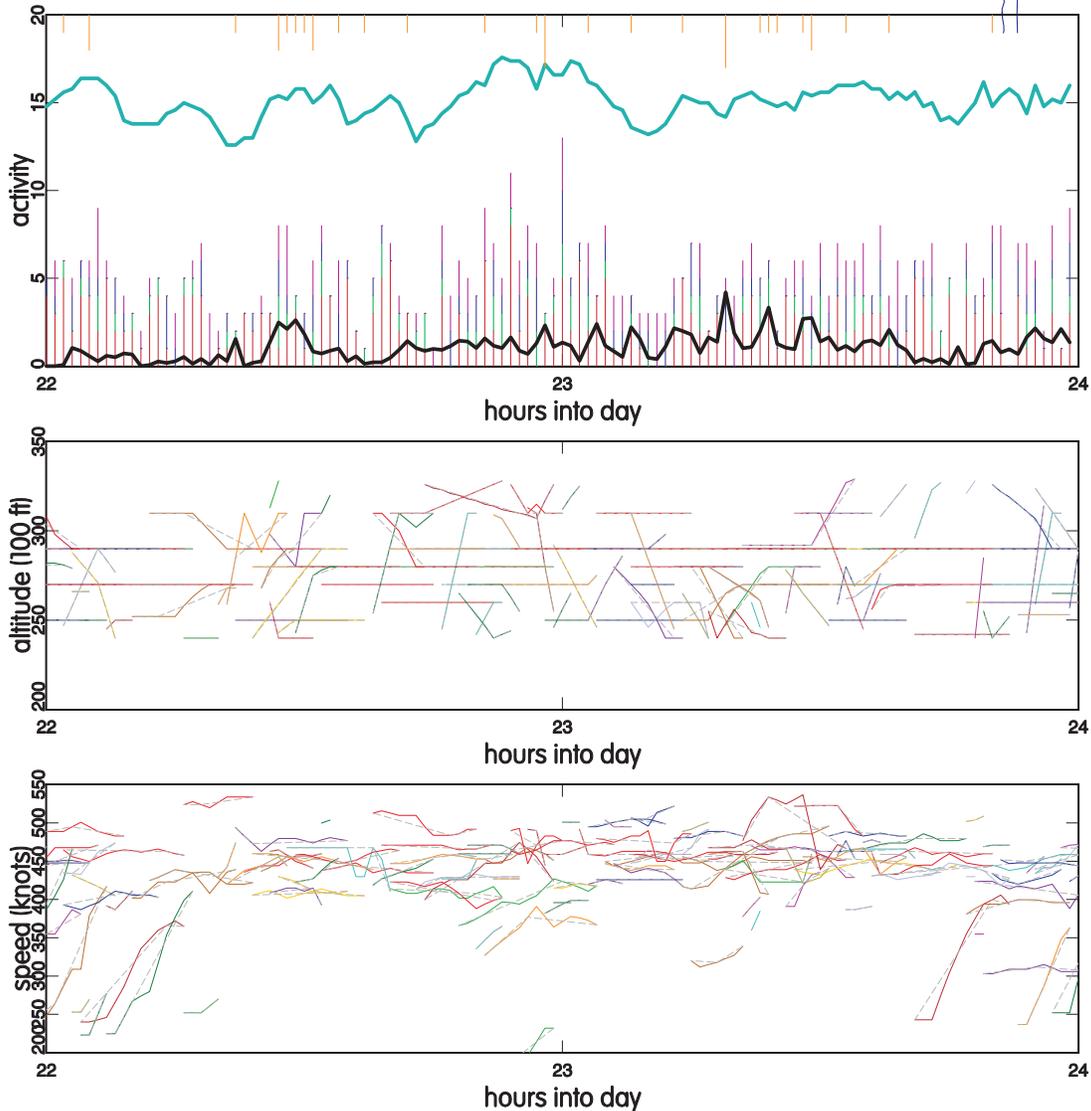
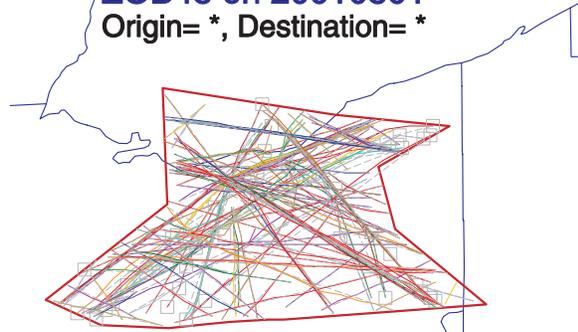


Fig. 4.7. Activities in ZOB48 for 2 hours on 1 August 2001. Colors for each flight are derived from a hash of the origin and destination.

0:00 to 24:00

complexity

Handoffs = 2399

Altitude Changes = 672

Speed Changes = 968

Vectoring Events = 1090

5-min Average Activity (+10)

Collision avoidances

Number in sample = 1294

ZOB48 on 20010801

Origin= *, Destination= *

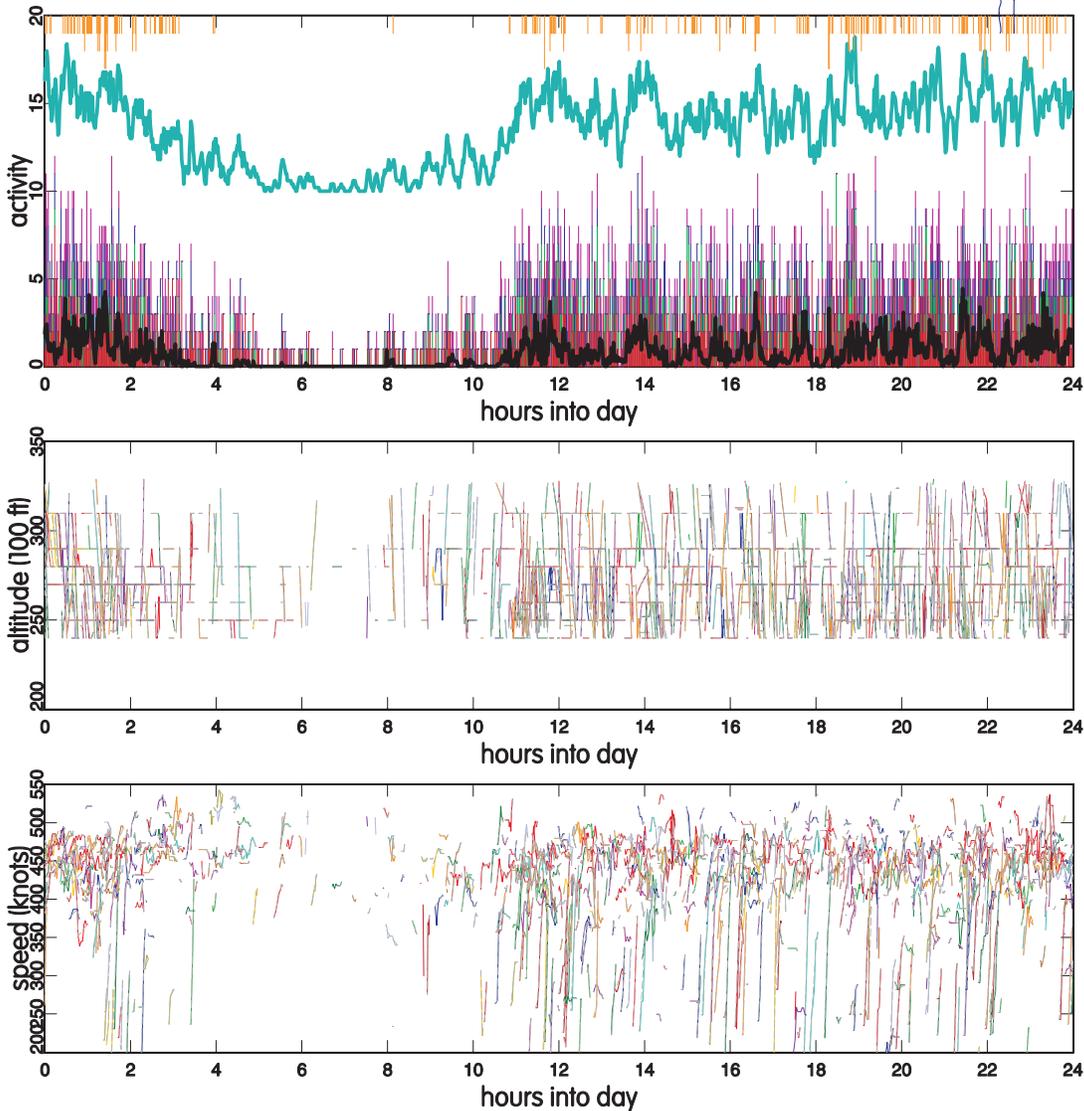
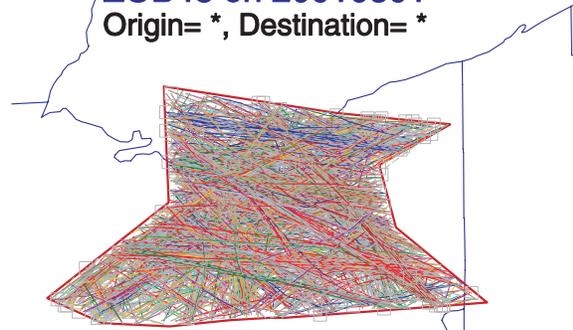


Fig. 4.8. Activity and complexity for the entire day of 1 August 2001.

Fig. 4.8 shows the activity and complexity for the entire day. A striking feature of this plot is that essentially the entire ZOB48 airspace is utilized. This makes it very difficult to hold flights in the sector if handoffs are missed or if adjacent sectors are closed suddenly. If we divide the complexity curve of Fig. 4.8 by the number of planes at each

minute and plot it versus the number of planes, we obtain the plot of Fig. 4.9. It appears that for this measure of complexity, the complexity per plane must decrease if there are large numbers of planes in the sector. This may indicate a limit to what a controller can manage using present techniques.

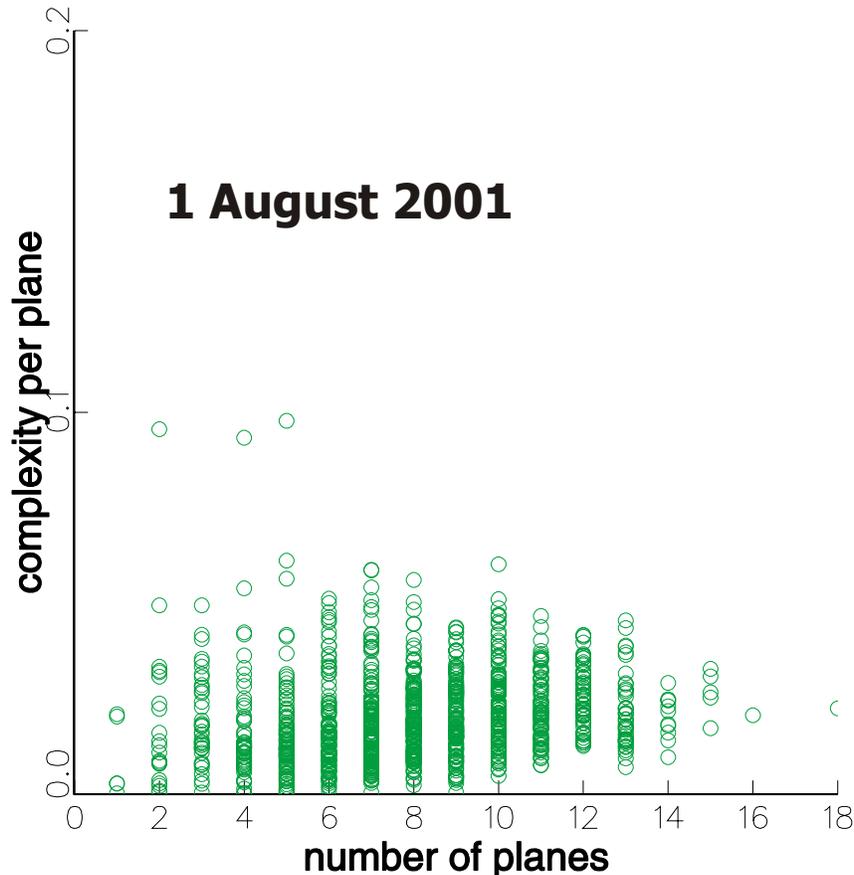


Fig. 4.9. The complexity of Fig. 4.8 is divided by the number of planes and plotted against the number of planes. When the number of planes in the sector is low, the complexity must also be low. However, when the number of planes in the sector is high, the complexity per plane also drops off, possibly indicating a controller limit.

Complexity as a function of time to spatial proximity

Another means of examining the complexity of a region of air space is the spatial proximity of flights that pass through it. Although controllers must maintain minimum separation, how close aircraft actually get to each other in relation to the minimum requirements is an indicator of complexity. In particular, the time until a pair of flights will become “close” is significant, for this captures the importance attached to deconflicting a pair of flights and anticipating the possibility of a conflict. For example, a pair of flights that will become close (however “close” is defined) in three minutes is much more of a concern than another pair of flights that will fly ten minutes with sufficient separation. Pairs of flights that never get close should contribute nothing to the complexity score.

We define a metric of complexity for each pair of aircraft flying at a particular time as the inverse of the time until the pair reaches some threshold of closeness. The value for each pair of active flights is summed to arrive at a metric for the time period. Closeness parameters may be varied when computing the metric, and the plots shown below use 1500 feet of altitude separation and 7.5 nautical miles of horizontal distance. ETMS data are used as input, and all ETMS data quality caveats apply.

Fig. 4.10 shows complexity as inverse time to proximity versus time in ZOB48 and a representative trio of its neighbors, ZOB27, ZOB40, and ZOB49 on 03 August 2001. ZOB27 lies to the north at the same altitude levels (it reaches higher to 34900 feet). ZOB40 and ZOB49 lie below and above ZOB48, respectively. The plots assume 1500 feet and 7.5 nautical miles of vertical and horizontal separation, respectively. Aircraft density (number of aircraft in the air space for each one-minute period) is also plotted for comparison.

Inverse time to proximity as a complexity metric shows the same trends as the activity metric above (Fig. 4.8). Peaks are still in the late evening, early morning, and early evening rushes. Early morning hours show no complexity at all, as to be expected. It is important to note the frequent oscillation between dramatic peaks and valleys in the ZOB48 complexity curve. This leads to two pertinent questions. Can metering of traffic smooth the curve? If so, will this allow an improvement in capacity? The answers are not obvious because controllers say they use the time between peaks to plan for the next traffic⁶.

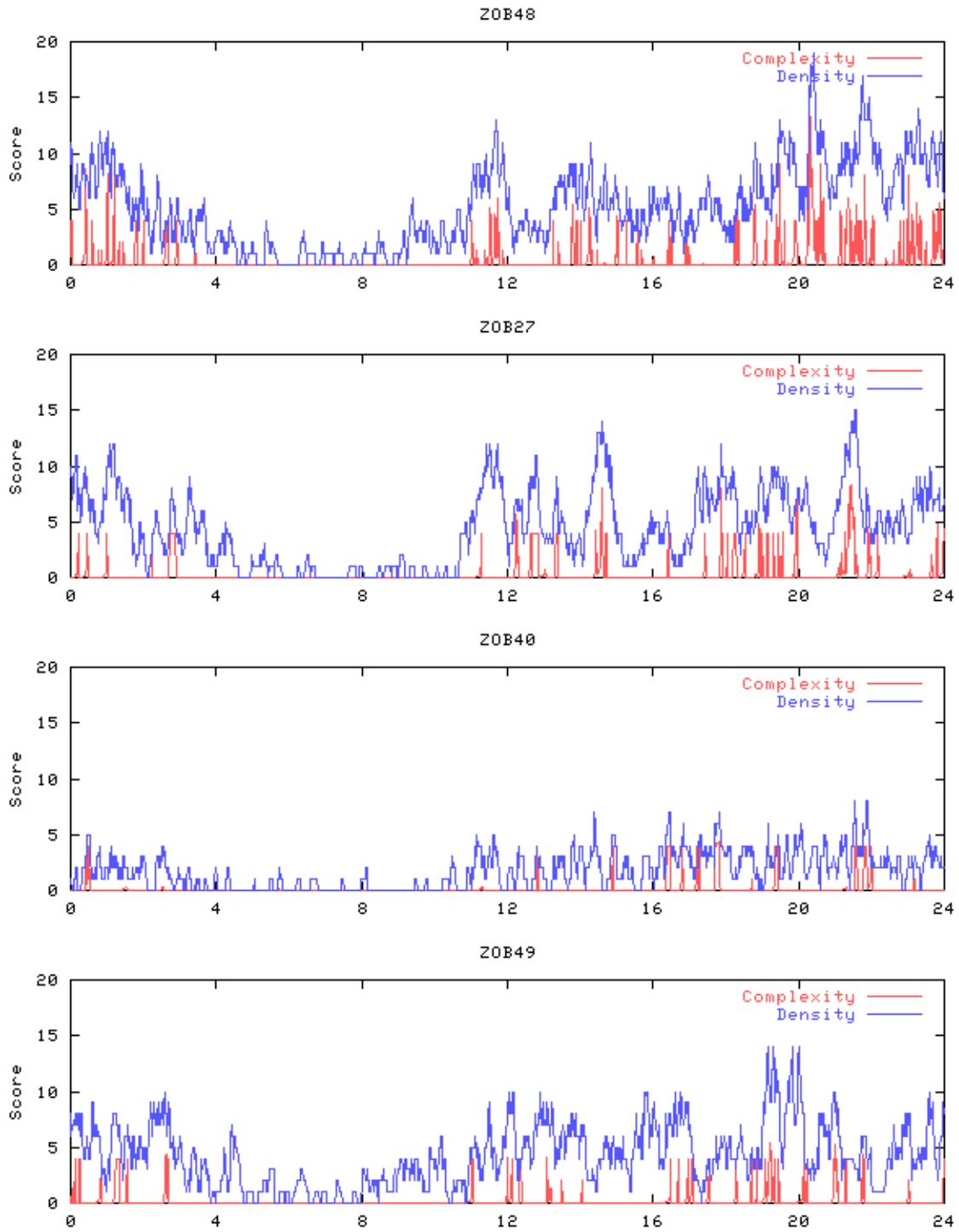


Fig. 4.10. Complexity as inverse time to proximity versus time on 03 August 2001.

4.5. Cause of ZOB48 complexity

Intuitively one may conclude that a reduction in traffic, or at least in the activity (e.g., altitude transitions) of the flights, would reduce ZOB48 complexity, and clearly a sufficient traffic reduction would do so. However, a comparison of ZOB48 and ZOB49 in Fig. 4.10 illustrates that complexity is not simply a function of density. Table 4-2 gives the sums of the inverse time to proximity scores (complexity) for each minute. These are the data plotted in Fig. 4.10. ZOB27's complexity is about two thirds of ZOB48's. ZOB49, which lies directly above ZOB48, shows little complexity (less than half of ZOB48) in spite of significant density. So, what is the source of ZOB48's complexity?

Table 4-2. Integrated complexity by sector on 3 August 2001.

Sector	Integrated Complexity
ZOB48	699.256
ZOB27	484.861
ZOB40	183.322
ZOB49	279.217

Cleveland Center traffic routes and flows, as seen in Fig. 4.1, indicate the difference between ZOB48 (Ravenna) and ZOB27 (Hudson). There are far fewer crossing flows in ZOB27 than ZOB48, which has no rival among all the sectors. This leads to the conclusion that ZOB48's complexity is due not only to the volume, density, and altitude activity of its traffic but results in large part from the inherently conflicting nature of the traffic, which, using present methods of traffic control, can only be solved by a change in route structure and direct routing policies.

We can reanalyze these complexity metrics for ZOB48 by creating a plot similar to Fig. 4.9. The results are shown in Fig. 4.11. Again it appears that if the plane density becomes too large, the complexity must somehow decrease so that the controller can handle the load.

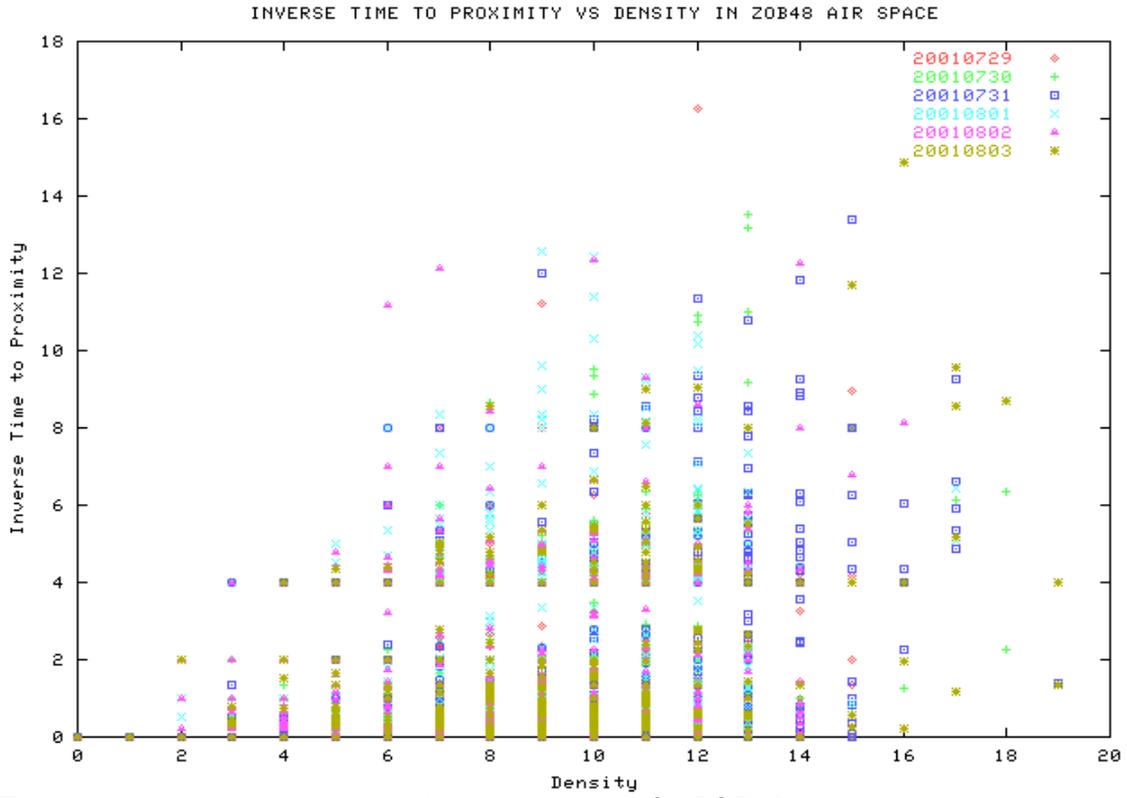


Fig. 4.11. Inverse time to proximity versus density for ZOB48.

5. New approaches to sector traffic management

The apparent limit to sector performance using present air traffic management techniques has been noted by the Royal Aeronautical Society as a driver for a new concept in air traffic management⁷:

The fundamental concepts of en-route air traffic management need to be changed. The historic dependence upon a ground controller to aircraft communication link should be progressively replaced by a system based upon allocated safe flight paths or “tubes of flight” of protected air space.

Getting air traffic through the airspace in a safe tube of flight is a matter of solving for the 4-D conflicts and finding a path through them. Several attempts to solve this problem have been made^{8,9,10}. We were especially impressed with the tube approach used in PHARE¹¹ and decided to develop our own 4-D conflict avoidance “look-ahead” scheme. An easy solution to this problem would facilitate solving the other problems of interest to us, namely

- Actively managing the traffic flows through a sector, i.e., simulating what a controller does, but at arbitrarily high data precision.
- Closing off regions of air space and seeing if the *same* traffic could still get through the restricted space.
- Seeing if more traffic could be accommodated in the same airspace at a busy time.

In addition, simulation of the traffic would allow us to generate higher resolution data based upon ETMS data, thus overcoming the data issues discussed in Section 2. We wanted especially to be able to present a proof-of-concept for all of these items using real data in the most challenging sector — ZOB48. However, we were constrained by time and resources from going much further than a proof-of-principle demonstration. Future work could explore a more realistic solution to these problems.

To achieve these goals we developed an improved approach to 4-D deconfliction that could allow pilots, air traffic controllers, and computer programs to easily find free paths in complex traffic situations. After describing this technique, we will use it to solve the above problems.

5.1. 4-D deconfliction

To achieve deconfliction, the planes can be regarded as occupying and excluding a region of space shaped like a hockey puck measuring 5 nm in radius and 2000 ft high (4000 ft high above 29000 ft). One plane cannot intersect the puck of another plane. The problem we posed is from the view point of the pilot of a plane: “If I assume that all other planes retain their present courses and speeds, what horizontal and vertical headings can I pick to avoid conflicts for a given look-ahead time?” We discovered a simple 2-D plot that will display this 4-D answer. The basic geometry is displayed and explained in Fig. 5.1. The key idea is that after the time a conflict occurs, it makes the region of airspace behind

it (on outer cylinders) inaccessible after that time. Therefore, we can project the conflicts with all of the planes onto a single cylinder that occurs at a later (“look-ahead”) time.

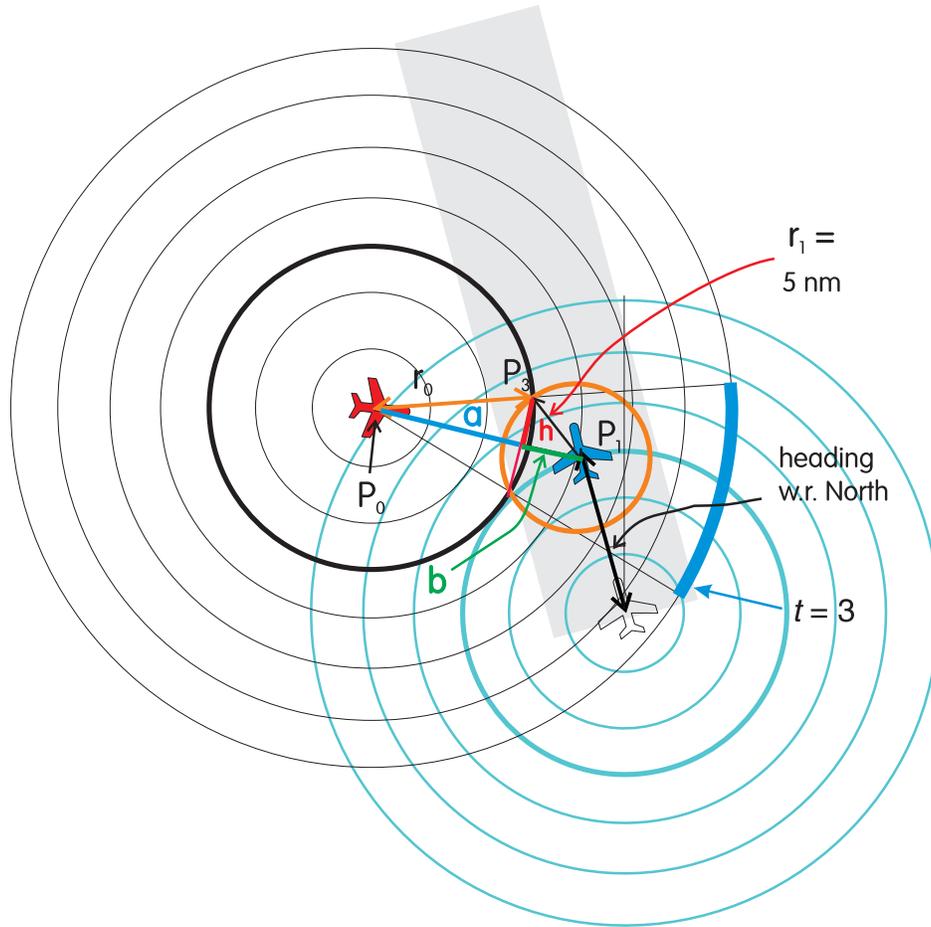


Fig. 5.1. The plane of interest is at P_0 (red), and the interaction with another (blue) plane (in the center of the blue circles) is shown. At $t = 0$; both planes are in the centers of their respective nested circles. Each circle going outward from the center has a radius equal to the distance the plane can fly in a time step. The blue plane is on a known path (heading speed, inclination). This plane’s conflict circle (5 nm radius) sweeps out the path shaded in gray. The situation at $t = 3$ is shown in the plot. The blue plane is then at P_1 , and must be avoided by P_0 at its $t = 3$. At this time it will be located somewhere on the darkened black circle, but it must be outside the Orange circle to avoid a conflict with the blue plane. The point P_3 is one of the two intersections of the orange circle with the darkened black circle. If P_0 avoids the interior of the orange circle, it will not conflict with the blue plane. However, if P_0 aims to miss this plane at $t = 3$, the region outside this (at larger radii) is also inaccessible to P_0 . Therefore, we can do all of the bookkeeping for collisions at a single circle for any (later) time, say $t = 7$ in this example. The thick blue arc is the region inaccessible to P_0 on this circle due to the situation at $t = 3$.

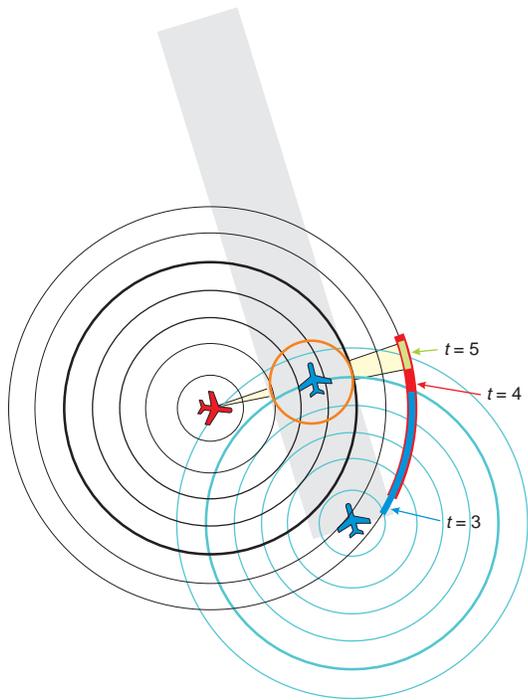


Fig. 5.2. Left, the situation at $t = 5$. As the blue plane moves from $t = 3$ to $t = 5$, its intersection with the red plane's position arc (at the same time) is projected onto the outer circle as shown. If the red plane stays outside of these arcs, it is impossible for a collision to occur. The side view of the projection is shown below. The height of the hockey puck is 2000 feet, but its projection becomes larger as shown below.

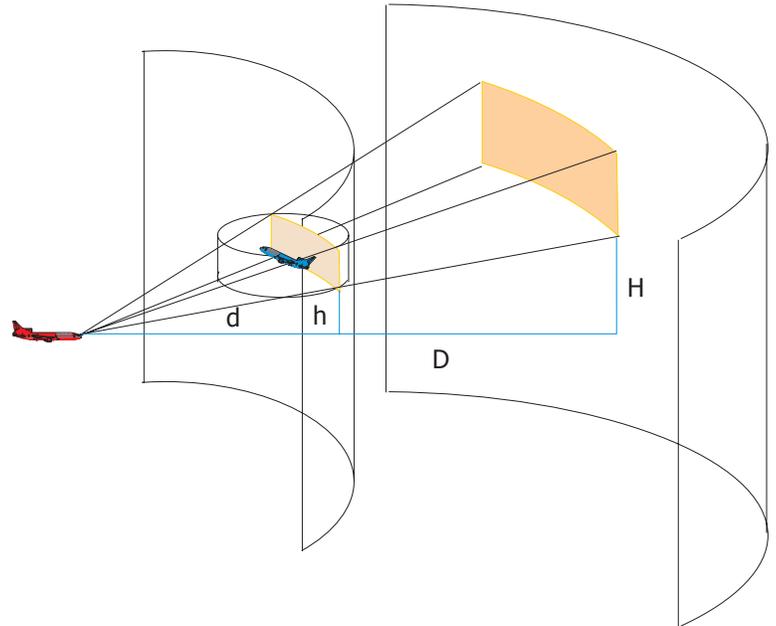


Fig. 5.2 shows how this projection works from above (at several times) and in an oblique view. The latter shows how intersections of each plane's hockey puck with the cylinder around the (red) plane of interest (at the time corresponding to the other plane) get projected to the outer bookkeeping cylinder. The height of this increases and may be above or below the red plane, depending on the relative altitude of the hockey puck. The projected area gets wider also, but because it is measured in degrees, its occlusion arc is unchanged. The region between the colored patches of Fig. 5.2 (right) is inaccessible to the red plane.

This same process is carried out for all other planes in the sector's airspace. More other planes will interact with this (red) plane as the look-ahead time is increased.

In an early version of the display, we colored the occluded region in Fig. 5.2 with a different color for each time step as is shown in Fig. 5.3. This plot is merely the unrolled outer cylinder in Fig. 5.2, and shows how each blob on the plot is made up out of a superposition of rectangles.

NWA1540

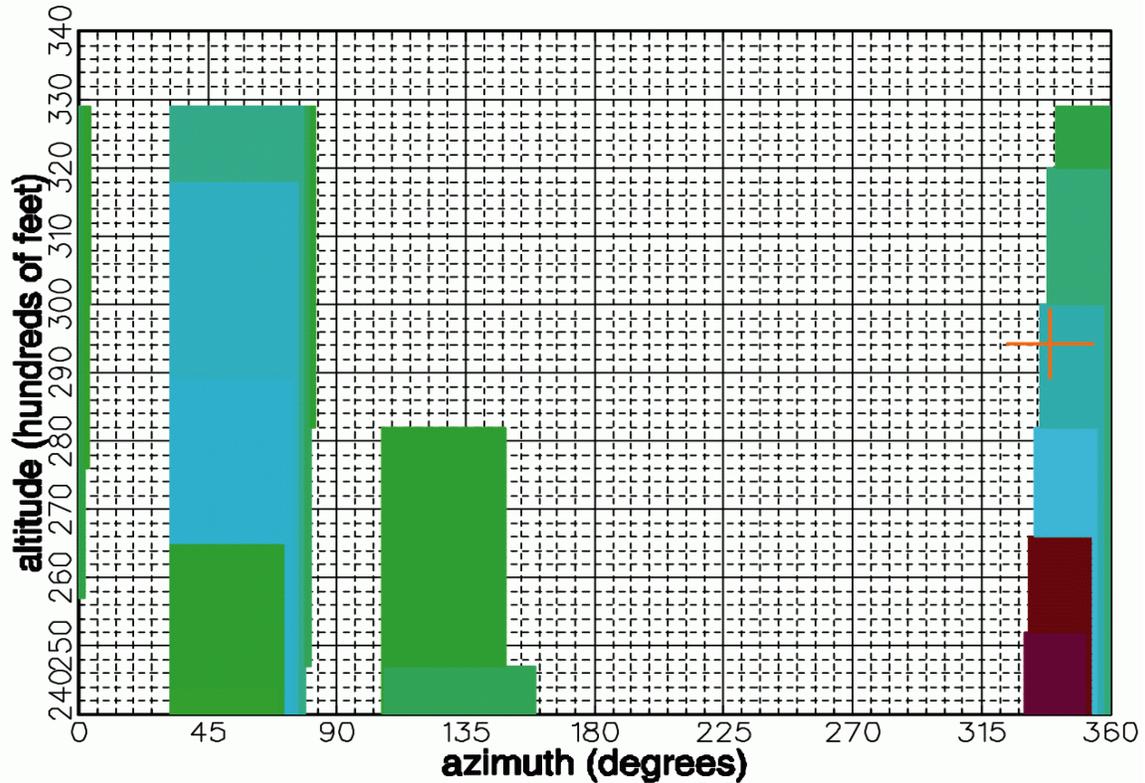


Fig. 5.3. An early version of the display with a different color used for each time step. This shows that the shapes in future similar plots are all made from the superposition of many rectangles.

The geometry to calculate the intersections of the two circles is quite simple¹². However, the intersecting cylinder analysis only considers the view from the top. It does not account for intersections with the top of the hockey puck space of the other plane. When the red plane is within 5 nm of another plane, we must occlude the projected space on the outer cylinder in a way that prevents the red plane of interest from moving vertically through the other plane. The geometry of the situation is shown in Fig. 5.4. Here, the red plane at the center of the previous black cylinders is located at (x_0, y_0) , a point that is above or below the hockey puck surrounding the blue plane located at (x_1, y_1) . The bearing of the plane at (x_0, y_0) is θ , and the bearing from the plane at (x_0, y_0) to the one at (x_1, y_1) is ϕ . The goal of this calculation is to calculate the horizontal distance to the edge of the hockey puck as a function of θ .

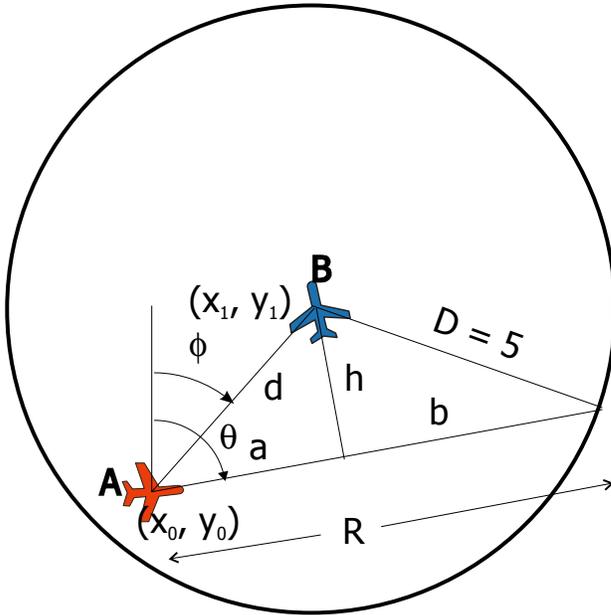


Fig. 5.4. Geometry when one plane is above the other.

the puck edge, at a heading of θ and at radius sN is given by using similar triangles

$$a_{\text{outer}} = a_0 + sN(a_0 - a_1)/R.$$

Here a_1 is 1000 ft (2000 ft above 29000 ft) above or below the altitude of the plane at (x_1, y_1) , B . There is some question as to whether the position of B should be allowed to move forward in time. We perform the above calculations for times at which the present position of A is above or below the hockey puck around future positions of B .

All of these calculations create forbidden regions on the outermost cylinder. We can unwrap the outer cylinder to obtain a 2-D plot of the 4-D results as a function of azimuth (0° is North and the angle increases clockwise) as shown in Fig. 5.5. A pilot or an air traffic controller looking at these pictures would have no problem determining the course and altitude heading that will avoid other traffic and get closest to his goal. In Fig. 5.5 AAL1052 interacts with 12 other planes and a closed region of airspace during the 10-minute look-ahead time. This is a very complicated situation for the unaided air traffic controller to handle, but using these displays, the pilot or controller can see just where to go in order to avoid conflict.

An example of the interaction between two planes is shown in Fig. 5.6. When planes get close to each other, the situation display can change rapidly as is shown in this example. Here AAL1052 passes on top of USA 299, and the code uses the algorithm explained in Fig. 5.4 to prevent the planes from changing altitudes (note the scalloped blue region in the bottom-right plot).

If all of the planes in the sector maintain their present course, speed, and climb rate, and if either AAL1052 or USA229 select a heading in a white area, it will travel on a route that is guaranteed to produce no conflicts for the next 10 minutes. If the look-ahead time

From the geometry of the two right triangles,

$$h/d = \sin(|\theta - \phi|)$$

$$a = (d^2 - h^2)^{1/2}$$

$$b = (D^2 - h^2)^{1/2}$$

$$R = a + b.$$

If the red plane, A , is on the top surface of the puck, it should not be able to descend, independent of angle. However, if A is above the puck, it can “peek over the edge” and descend slightly. The radius of the projection cylinder is given by the distance the plane of interest can fly in the look-ahead time, sN , where N is the number of time steps we look ahead and s is the distance flown per step. The altitude of this line that is tangent to

is increased to the time to cross the sector, this method can select a “pipe through space” for the plane that is guaranteed to get it through the sector with no interference if the other planes stay in their pipes, as we will show later. If the destination spot is not free (white) at the time the plane enters the sector, he can select a new destination point, or change his speed (which changes the plot). The plots extend to all altitudes because planes that climb or descend into the sector that are not at the edge of the sector must aim above or below the sector altitude limits.

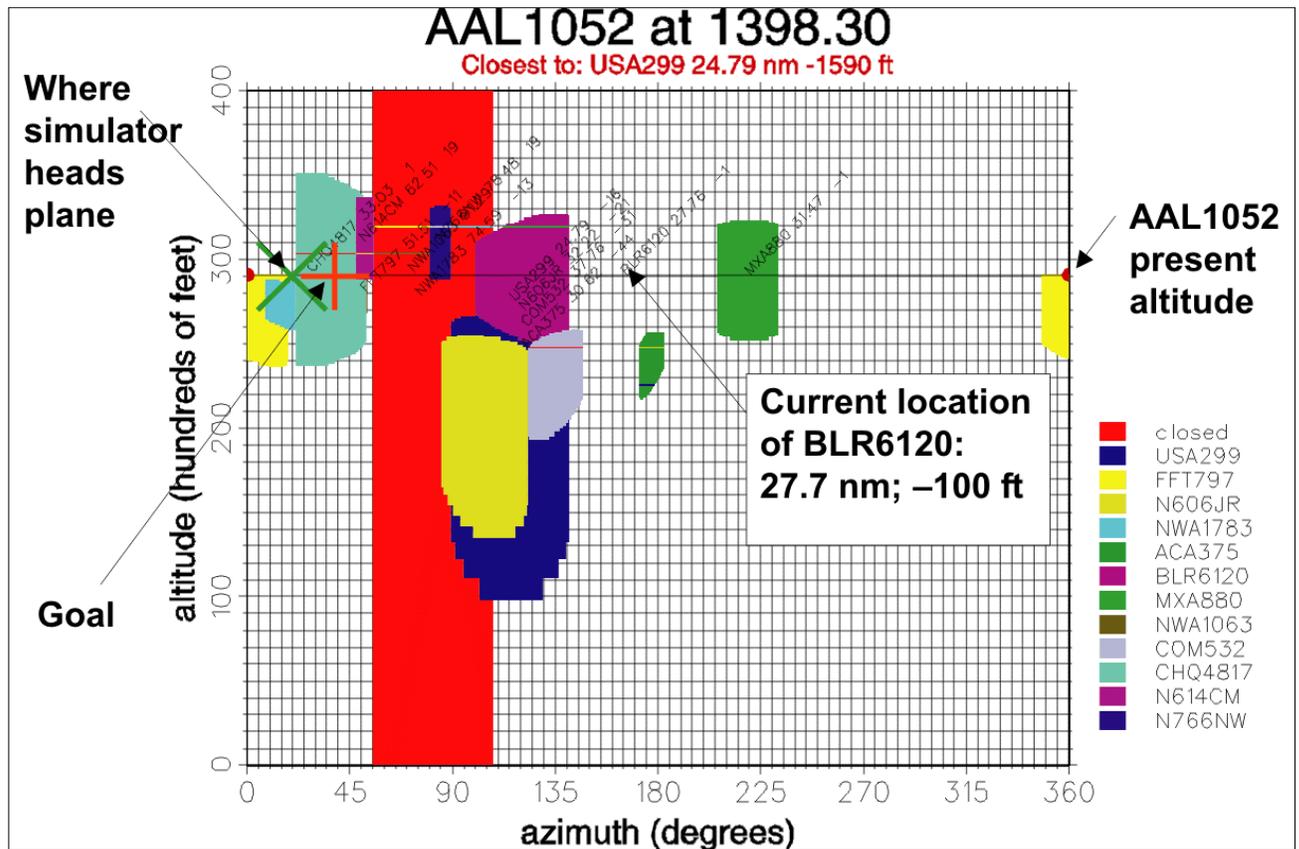


Fig. 5.5. Annotated version of the 4-D conflict avoidance plot in an example with a closed region of airspace.

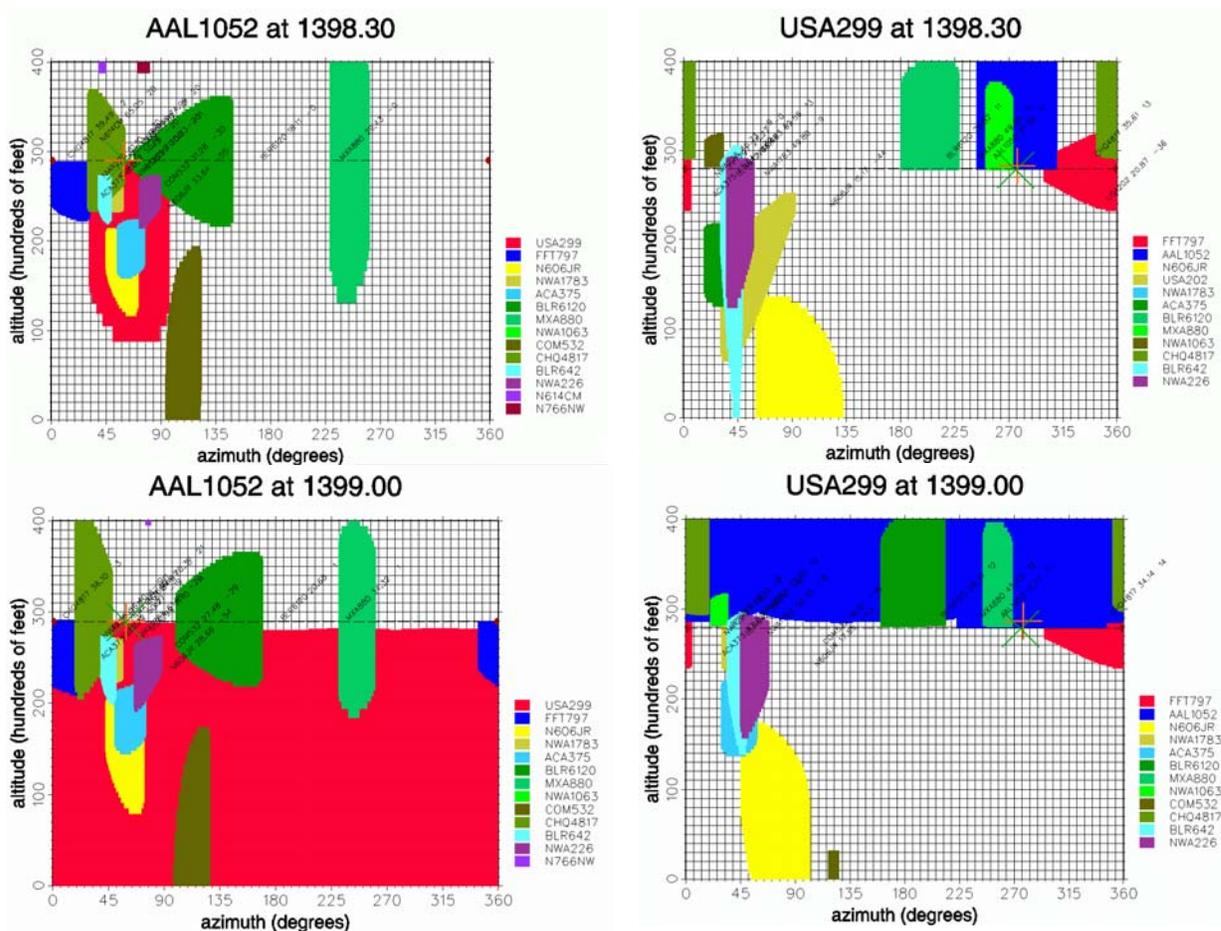


Fig. 5.6. The interaction between AAL1052 and USA299 is shown at two times 0.7 minutes apart. In each plot, the orange + sign is the heading towards the (predetermined) exit point from the sector. If it does not occur in a white part of the plot, a conflict will occur. The Green X is where the traffic management simulator put the plane on its next time step. The horizontal line with red semicircles at the ends is the current altitude of the subject plane. These planes are passing over each other 1000 feet apart. When the planes are within 5 nm of each other, the patch for the other plane fills a half space, which prevents the other plane from changing altitudes. The projected patches for some of these planes are not rectangles because they are changing altitudes. When this occurs, the patch at each time step is displaced vertically from that at the previous step. The text written on the plots is the aircraft id (acid) for each of the planes in the legend at the time of the plot, together with the distance from the subject plane and the altitude difference. These locations are often not at the color patch for the plane because no interference occurs at the current position.

5.2. Assumptions

Due to the exigencies of the task schedule, we made certain assumptions in applying our 4-D deconfliction technique. Some of them make it harder to solve the sector traffic problem; others are somewhat unrealistic. However, we believe all of the assumptions could be eliminated with just a little more work.

Traffic based upon actual ZOB48 data

To be realistic, we used actual traffic in ZOB48, and we concentrated on the evening rush from 22:00 to 24:00 on 1 August 2001. By actual traffic, we mean that we use the interpolated times and locations for the entrance and exit of each flight through the sector, assuming that the “handoff” occurs at the physical sector boundary. Keeping the exit time and location prevents disrupting other sectors. However, it also prevents cost savings that could accrue because of more direct routes. The simulation adjusts the plane’s speed to try and meet the exit time. Because of problems in the ETMS data, it is possible that two planes enter the sector closer than the separation limits allow. If this occurs, we delay the entrance of the second of these planes until the interference is over.

Ascent, descent, and speed

A realistic simulation would have climbing planes climb as soon as possible, and descending planes descend as late as possible. We did not do this to make things a bit simpler; our flights climb or descend linearly (when not obstructed). We do, however try to restrict ascent and descent rates and speeds to reasonable values. We did not retain the aircraft types in the data sets, so used 2500 ft/min for the maximum ascent and descent rates. Medium jets can climb at up to 4000 ft/min, and heavies up to 3000 ft/min. Descent rates are limited because powered descents build up the aircraft’s speed too much. For each plane, we found the actual maximum and minimum speeds in the sector and used those as limits. We have implemented the change in altitude separation to 2000 ft above 29000 feet in altitude; it made little difference in the results.

5.3. Simulation

We believe that our deconfliction plot (Fig. 5.6) makes it very easy for a pilot or a controller to select the best heading and ascent/descent rate. However, what is obvious to the human eye is often a lot harder to convert into a computer algorithm. Reference 9 nicely defines the goal of an air traffic management system:

Air Traffic Management (ATM) is based around one major issue: keeping aircraft apart. To this ‘anticollision’ function an ideal ATM system will add, in the learnt by rote phrase, the “safe, economic, orderly and expeditious” operation of the aircraft. The aircraft should be safe, although there is no real definition of what ‘safe’ means only standard separation definitions. The economic operation of the aircraft should mean as far as is possible giving the aircraft operator or pilot the flight-path that has been requested. However, to be ‘orderly’ the separation should not be achieved with a flight-path made up of repeated short term deconfliction manoeuvres and as far as is possible there should be no delays to the aircraft’s flight to its destination.

Given our assumption list, we tried to implement these goals in our simulation. Orderly turned out to be the hardest goal to achieve. We did not calculate the economic impacts here, and as pointed out above, our climb/descend algorithm is definitely not optimum (but could be easily changed). Here we list the key ingredients of our computational recipe.

Where to go on the next step?

The algorithm has two phases: goal seeking and conflict avoidance. We know where the goal is, but if the plane suddenly passes an obstacle and heads directly back to the goal, the motion can become non-orderly. The code picks a desired goal point and does a spiral search around this point to find a non-obstructed heading for the plane's next step. Originally, we chose the goal as the start for this search, but sometimes the search would find a point that was not between the plane and its goal. Accordingly, we moved the start point for the search to halfway between the plane and its goal, which cured the problem.

The dance of death

The code updates the position of one plane at a time. When planes are close to each other, the interference plot can change dramatically with time (see Fig. 5.6). A plane that is blocked from its goal changes heading, so the other close plane then must move from its goal; it moves and at the next step, the first plane heads back to its goal. This process of switching from avoidance to goal seeking prevents the planes from actually avoiding each other, and also it creates a very non-orderly solution. To solve this, when planes are within 10 nm and the minimum altitude separation, we keep them on their present heading (if it is clear!) until they are past the other plane.

Breathing room

We can add an extra allowance for error in the collision avoidance algorithm. Making the hockey pucks bigger creates less room in the airspace. We used a radius of 6 nm and altitude separation of 1200 ft at altitudes of 29,000 ft and below, and a vertical separation of 2100 ft above 29000 ft. In addition, after the code finds a solution, it checks to see how close it is to an obstructed region. If there is free space perpendicular to the obstacle, the solution point is moved somewhat away from the obstacle. We tried to move the planes 1.25° in heading and 1000 ft in altitude. Note however, that the altitude heading is the change at the look-ahead time — 10 minutes into the future.

Results

The simulation program seems to produce safe, orderly and expeditious routing through ZOB48, even at rush hour. The simulation code produces data sets of each flight's simulated course, altitude and speed, and also a movie-like replay of the sector events. In addition, it actually produces shorter routes through the sector. Fig. 5.7 shows the ratio of the simulated distance flown to the actual distance flown in the sector. Almost all flights manage to fly a shorter path through the sector using the simulation techniques. The lower lobe of the distribution is due to planes that originally made a dogleg through the sector, but that fly on a (shorter) straight path in the simulation.

simulation on 20010801 from 0:00 to 24:00

total number of flights = 1068

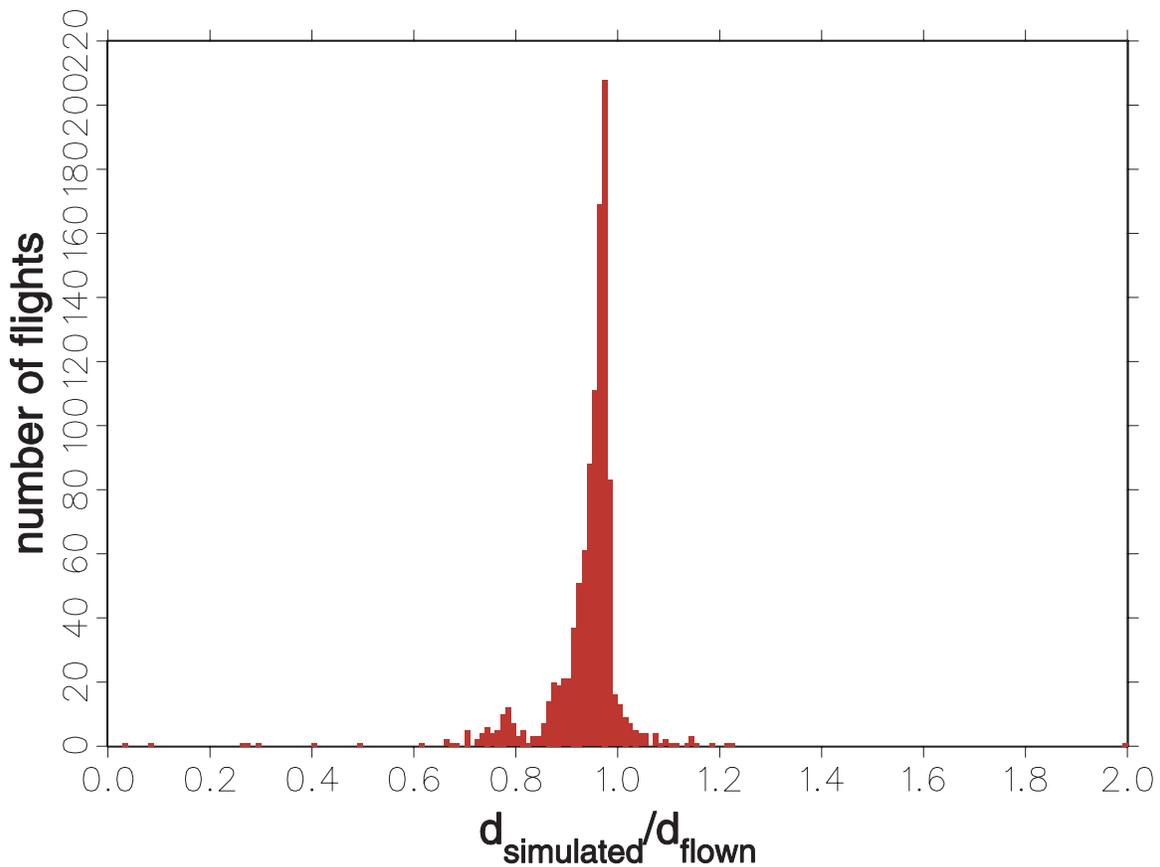


Fig. 5.7. The ratio of simulated distance flown in ZOB48 to the ETMS (actual) distance flown. The data points with small ratios are from the flights part-way through the sector when the simulation starts.

5.4. Restricted airspace

With the ability to successfully simulate and “control” sector traffic, it is possible to see how this would change when part of the airspace is closed. Here we restricted ourselves to areas smaller than a sector. Examples of such closure might be a particular configuration of SUA or weather cells. To simulate closed airspace in a quick manner, we decided to place a cylinder in the middle of ZOB48 that we treated like another aircraft. However, in this case, the speed is zero, and the altitude covers the ground up to 40,000 ft. The introduction of the obstacle changes several things in the simulation code.

Entrance/exit positions

Climbing and descending traffic can cross the ZOB48 boundary inside or very close to the obstacle. If a flight’s start or end was within 10 nm of the obstacle, it was moved. In each case, the point was moved to be 10 nm outside the obstacle, and on a line perpendicular to the line between the obstacle and the goal or origin.

Obstacle avoidance and traffic flow

Whenever a straight line from the plane to its goal intersects the obstacle, we temporarily change the goal so that it is on a line that goes through the obstacle and is perpendicular to the line between the plane and its actual goal. For planes going around the obstacle counterclockwise, we make this new goal 15 nm outside of the obstacle; for clockwise traffic it is 10 nm. This temporary goal is updated at each time step, so the traffic flows nicely around the obstacle. The direction that a given plane takes to avoid the obstacle is determined by the shorter way around. If we made all traffic flow around the goal in the same direction, the result would probably be more orderly, but require longer flight distances for many planes.

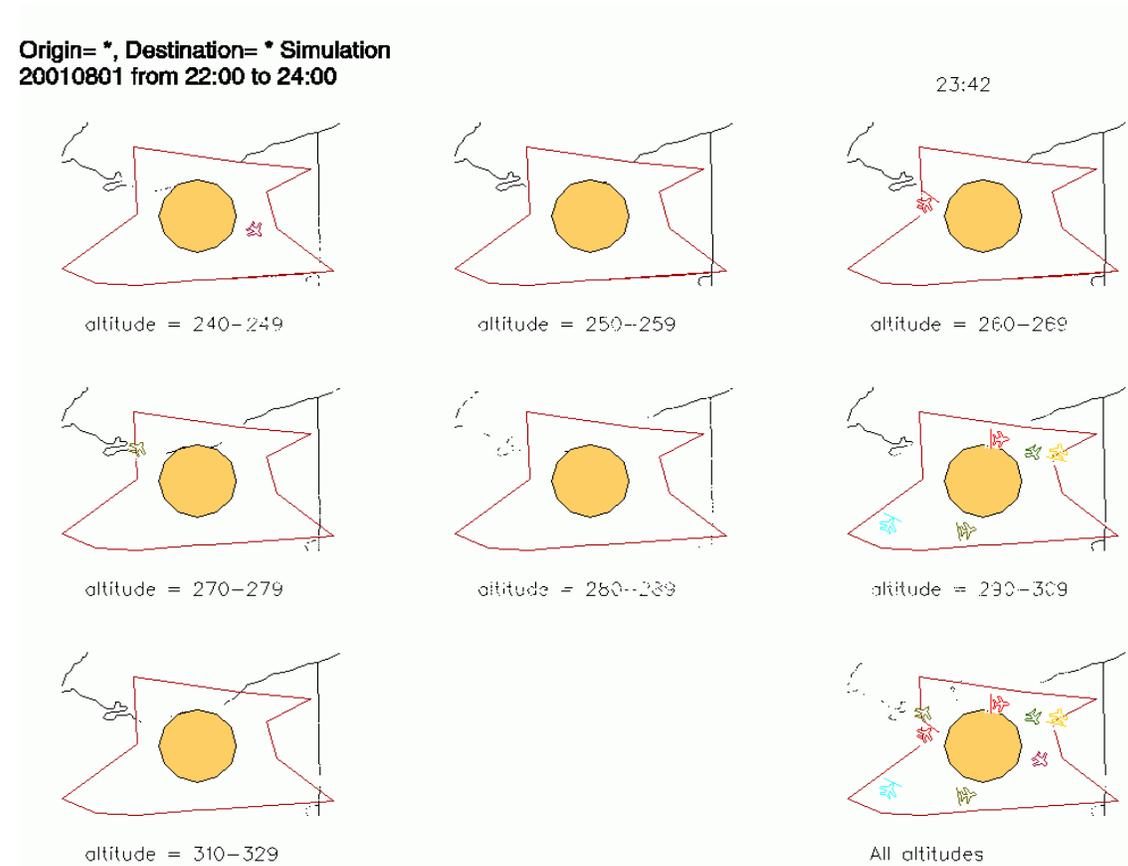


Fig. 5.8. Scene from a video clip of the planes circumventing a 20-nm radius obstacle placed in the geographic center of ZOB48. The lower right plot shows the planes at all altitudes.

The simulator is able to get all the traffic through the sector with either a 10- or 20-nm radius obstacle in the center shown in Fig. 5.8. The simulator changes the speed of the planes to maintain their exit times, provided the flight's entrance or exit point was not within the obstacle. Fig. 5.9 shows how two flights interact with each other while avoiding the obstacle. Fig. 5.10 shows a top view of flight tracks with and without the obstacle.

These tracks are for flights that interact (i.e., have a patch in the azimuth vs. altitude plots) with USA299.

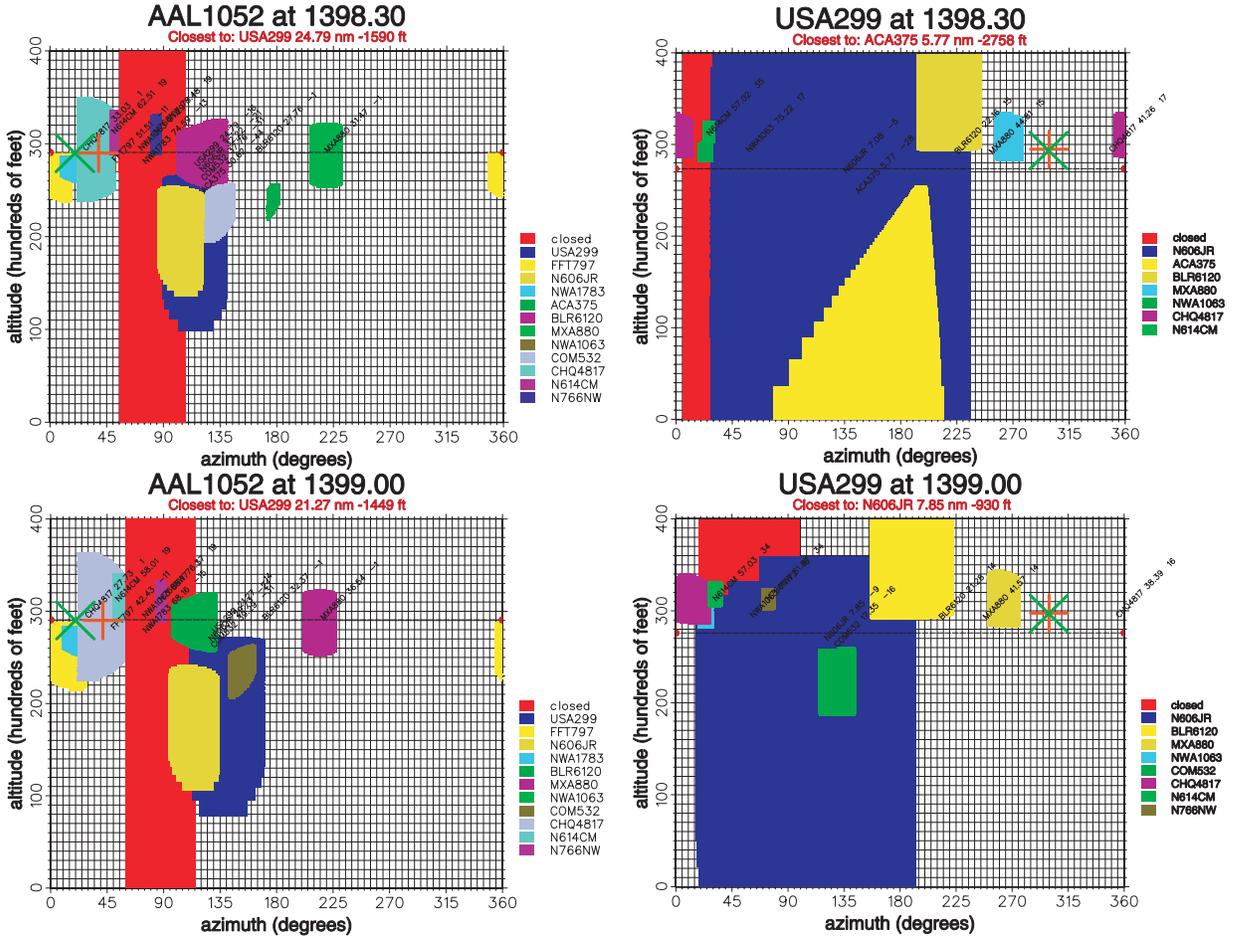


Fig. 5.9. Here is the same situation as is shown in Fig. 5.6, but with a 20-nm radius obstacle (shown in red). Avoiding the obstacle has caused AAL1052 to veer away from USA299, and the conflict between the flights has been eliminated. This is also shown in Fig. 5.10. AAL1052 is further from the obstacle than it would like to be in order to avoid CHQ4817. Notice how rapidly the plot can change when flights are close together as is the case for USA299.

The simulator calculates this interference plot for each plane, and at each time step to determine how to proceed. Nonetheless, the simulation only takes about 1 second per minute of real time on a 1 GHz Pentium III PC. The performance of the simulator with a 20-nm radius obstacle is shown in Fig 5.11. To make a fair comparison, we ignored flights whose start or endpoints were within the obstacle (and hence were moved), and also those that went outside of the sector's bounding rectangle. In general, the center of the distribution is centered close to 1.0, which means that the average flight will take the same time to get through the sector as before with no obstacle. The bimodal distribution in both versions of this plot is due to the fact that a group of planes ordinarily takes a dogleg route through the sector. Vectoring usually occurs near the Jetway intersection West of Cleveland. The simulator usually succeeds in finding a straighter path.

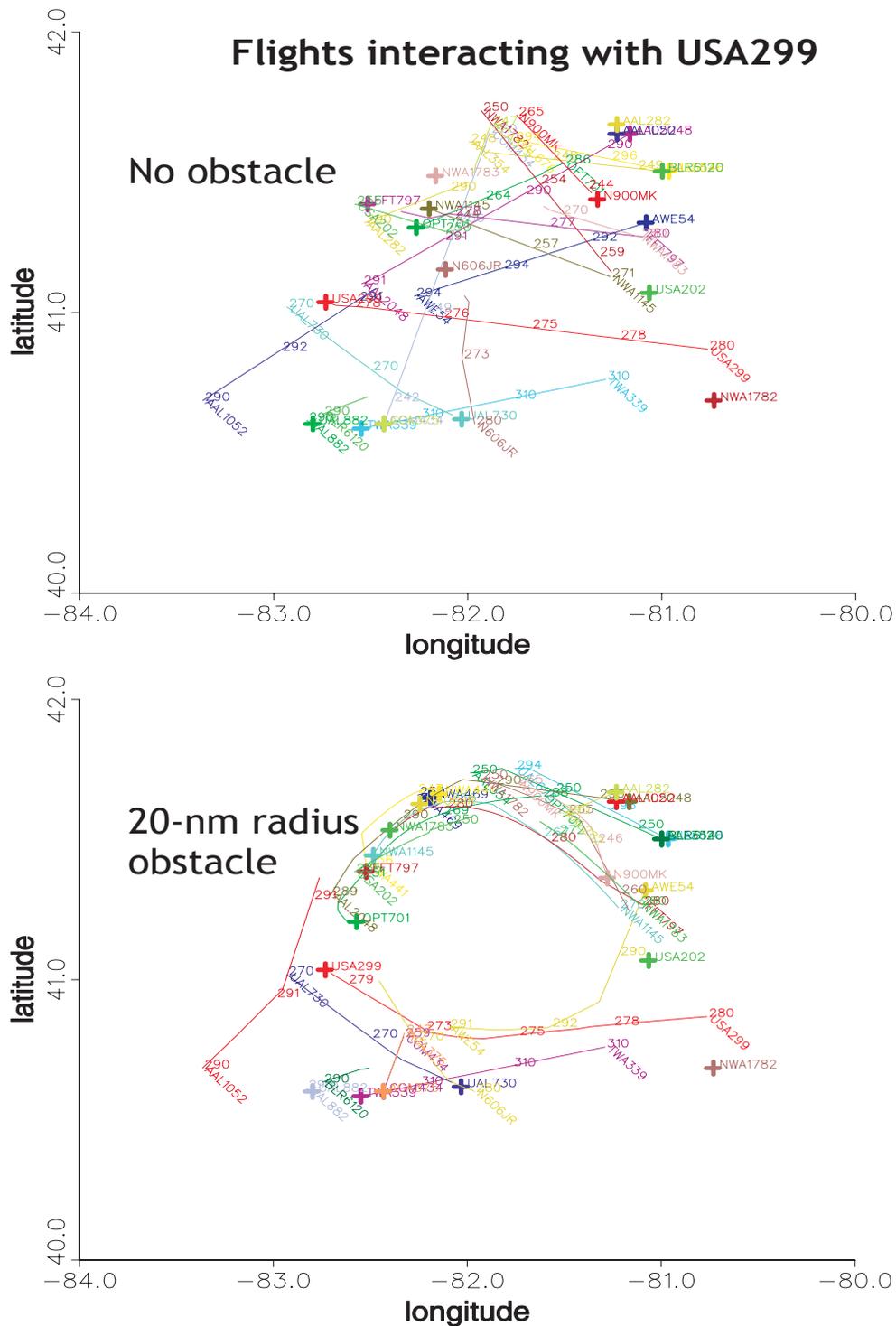


Fig. 5.10. Flights interacting with USA299 (dark red) with and without a 20-nm radius forbidden region. The altitude (hundreds of feet) is shown every 30 simulation steps. The + sign is the goal of each flight. Not all flights reach their goal here because the path is stopped when USA299 leaves the sector. Data are for 1 August 2001 around 14:00 GMT.

simulation on 20010801 from 0:00 to 24:00 with 20 nm obstacle

total number of flights = 581

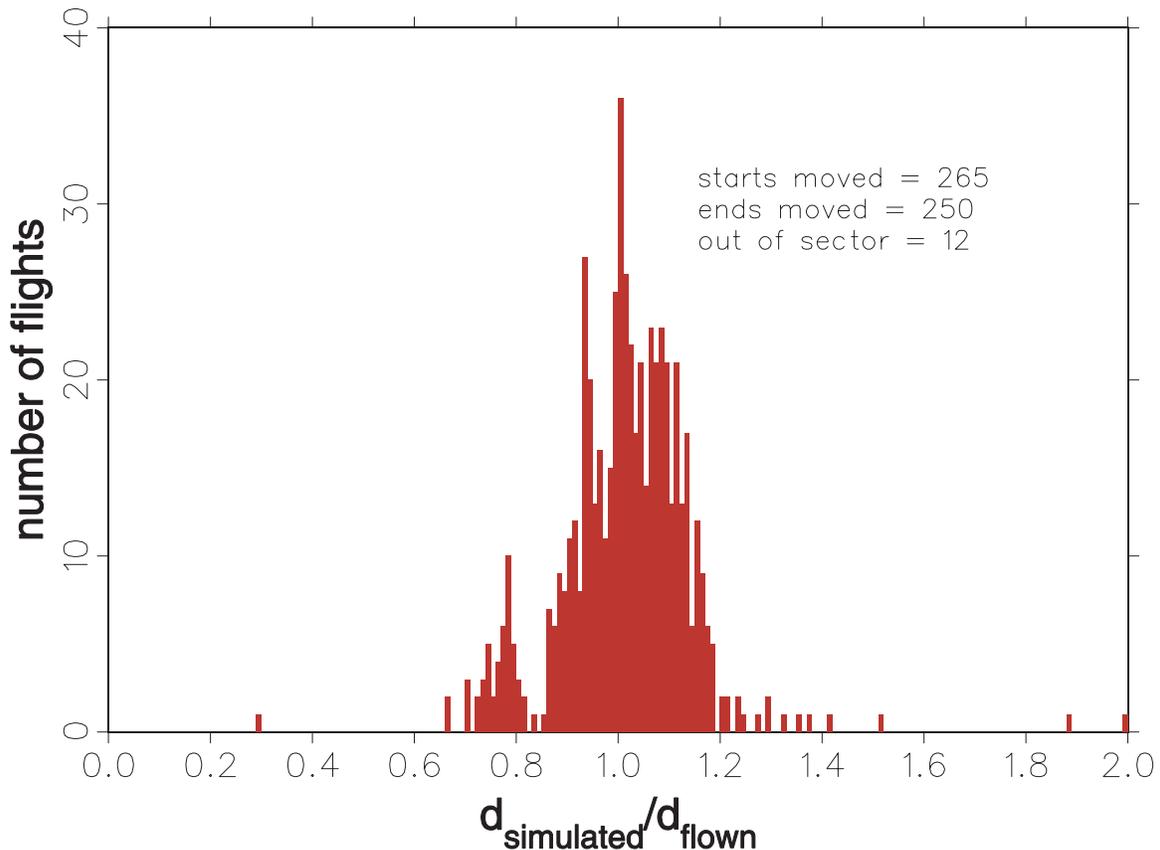


Fig 5.11. A comparison with Fig. 5.7 for the case with an obstacle. To make a fair comparison, we had to eliminate those flights that had their endpoints moved (to be outside the obstacle) or else that went outside the sector's bounding box (in which case the code stops following them). This result shows the same bimodal distribution as Fig. 5.7, but it is broader and centered about 1.0. The outliers near 2.0 are from flights that the code deemed to miss their target. They got close, turned around, and found it.

5.5. Extra planes

In order to see whether we could fit extra planes into the ZOB48 airspace at its busy time, we rewrote the code to reserve airspace (i.e., a pipe) for each plane that would keep it collision free from start to exit. We only used single straight pipes, which is not totally realistic, but in fact makes less efficient use of the airspace. This assumption could easily be eliminated with some additional work. Once all of the actual ZOB48 traffic is assigned to its pipe, we can then add extra pipes representing flights between city pairs.

Because the simulation goes forward in time, we decided to start looking for a pipe for a plane 2 minutes ahead of the actual sector entrance time. The performance of the pipe finder for 4 days is excellent as shown in Fig. 5.12.

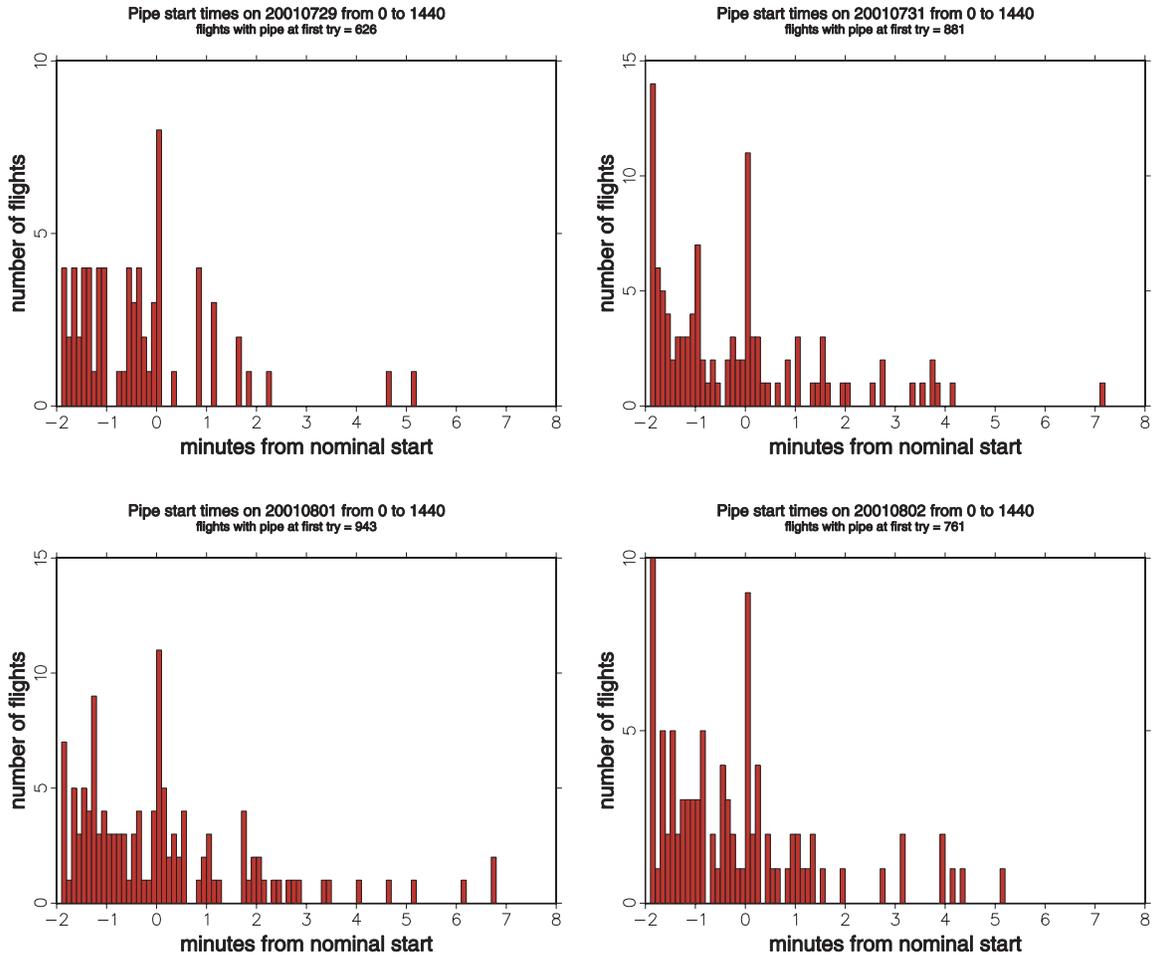


Fig. 5.12. The vast majority of flights find a straight pipe on the first time step as indicated in the caption. The spike at 0 minutes is due to flights in the air when the simulation starts. If jointed pipes were allowed, more pipes would be available.

When we look for a pipe, we aim in a straight line to the goal. In this version of the pipe finder, we allow an angular deviation from the desired goal that keeps the exit pipe within 10 nm of the original exit point. We also allow the altitude of the pipe to change by up to 4000 ft, provided that the flight does not exit the top or the bottom of the sector. The simulation for an entire day takes less than 4 minutes on a 1 GHz Pentium III PC. Most of this time is spent finding extra pipes and writing the time to the screen.

In Fig. 5.12, the planes that are not accommodated promptly (a few dozen) would be able to take jointed pipes if they were available in our simulation; we do not envision planes “waiting around” near the sector boundary for a pipe. In a real system, they would be reserved on a more global basis, and would be available when a plane arrives.

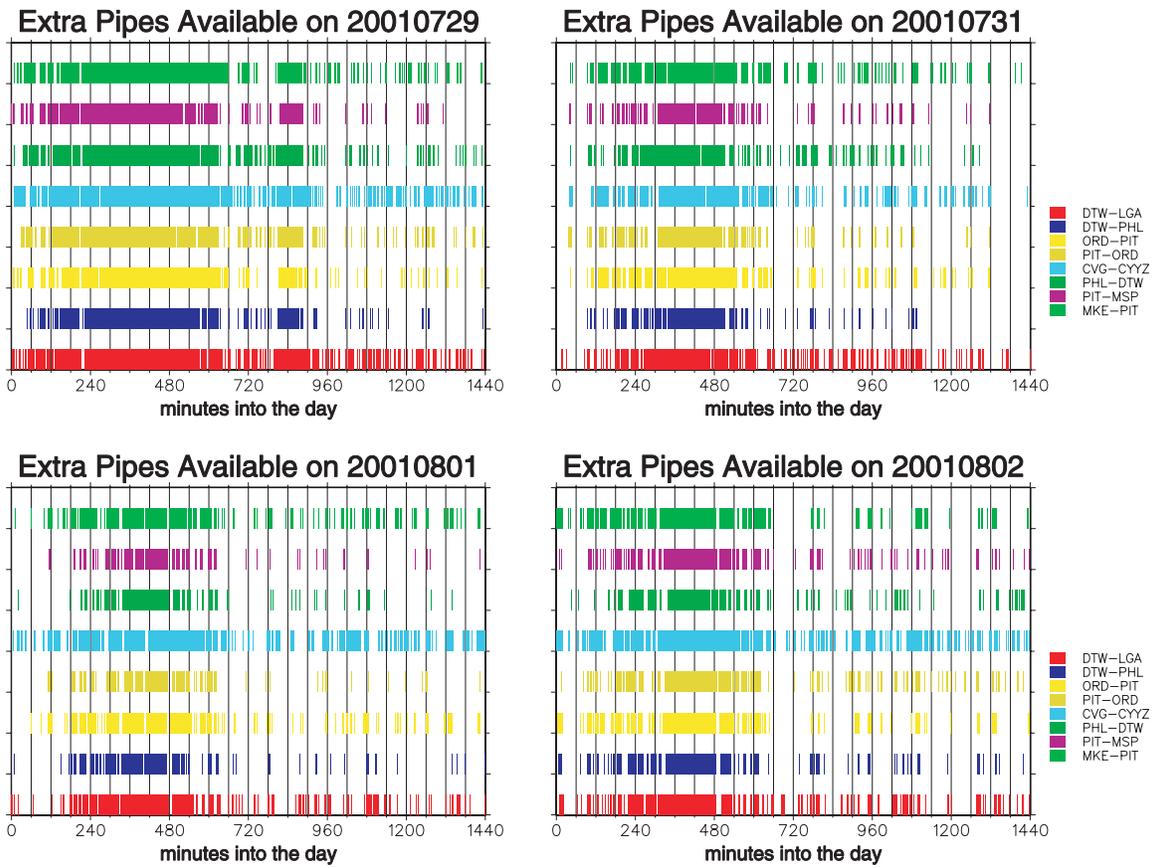


Fig. 5.13. Extra pipes between common ZOB48 city pairs. If a pipe was available at any time step in a minute, the whole minute was marked as available. July 29 was a Sunday, so more extra pipes were available. In ZOB48, Tuesday–Thursday are generally the busiest days.

Once the scheduled flights have their pipes, we can see if we can find extra pipes at each time step for flights between popular destinations (as an example). The calculation is inconsistent because we treat each of these extra pipes as a separate entity. We do not reserve the airspace taken by these extra pipes, so there must be a time delay between flights on these new pipes to avoid any interference among them. The result of this calculation is shown in Fig. 5.8. As expected, it is hardest to get an extra pipe during the morning and evening rush hours. If jointed pipes were allowed, the climb and descend phases would be more realistic, and it would be easier to find available pipes. Using jointed pipes would be a useful topic for a follow-on study.

5.6. Application of these techniques to ATM

How do we envision these techniques being applied to assist in the current ATM scenario? The key point to understand is that if a flight stays on the course determined by avoiding conflicts in the 4-D deconfliction display, then it will remain conflict-free until the look-ahead time, provided all other flights remain on their assigned conflict-free paths. Therefore, ATM should be able to create a conflict-free path through the sector upon acceptance of a flight.

We envision the following scenario:

- A flight appears on the usual 2-D radar display and is accepted by the controller.
- He/she clicks on the flight's icon and our 4-D deconfliction display for that flight appears. However, the look-ahead time needs to be adjusted, depending upon the flight:
 - The time through the sector is used for straight-thru flights
 - The time to reach altitude is used for ascending flights
 - The time to reach start of descent is used for descending flights
- A clear goal at the look-ahead time is selected for each flight on the 4-D deconfliction plot.
- For ascending/descending flights, a second pipe is created from the first goal to the sector exit.
- If necessary (for example to avoid restricted airspace) extra joints can be added to the pipes.
- The airspace in the selected path is "reserved" in the computer code, and will appear as a conflict for all future pipe selection attempts.

This scenario should significantly reduce controller-plane interactions. It can also serve as the essential element in a more global planning approach wherein the user, in collaboration with ATCSCC, might preselect pipes for the entire flight.

6. Conclusions and recommendations

In this study, we have shown that the airspace can be characterized as both a large-scale entity and a small scale one. There is an obvious definite correlation between the weather and the sector load, and we can detect it by looking at the TZ density and comparing it to a running average.

We concentrated on ZOB48 in the Cleveland Center. It is characterized by complicated traffic patterns, climbing, descending, and crossing traffic streams, and by a large number of planes. The complexity in ZOB48 was characterized, and it appears that there is a tradeoff between the number of planes that the sector can handle and the complexity of the traffic patterns.

We have showed that by using a better 4-D deconfliction display, one can

1. Simulate traffic management in the sector,
2. Maintain the traffic load even when a significant portion of the sector is closed, and
3. Add extra flights through the sector at many times in the day.

We believe that traffic flow management could be maintained and improved by applying our techniques at the control console, in the plane cockpits, and in a more global planning system. The ability to quickly find and assign pipes to flights seems to have many advantages, notably greatly reduced demands on sector controllers.

However, in this limited effort, we have certainly not built a new air traffic control system; we have only given a proof-of-principle of the benefit of using advanced 4-D deconfliction techniques and tools in congested air space. Missing are features such as

- Implementation of “climb as soon as possible, descend as late as possible,”
- Closer consideration of the performance capabilities of individual aircraft types.
- Changing a pipe due to turbulence or changing weather conditions,
- Pipe selection negotiation (between controller and flight or airline operating company), and
- Optimizations due to wind patterns, proper climb and descent rates.

None of these limitations are impossible to overcome, and the team recommends addressing these issues in a future study to enhance the current work.

Appendix A.

Data Cleaning problems on 1 August 2001

These incorrectly identified separation violations are an indication of data cleaning problems rather than actual incidents or controller mistakes. A flight “owns” ± 200 ft about its altitude level, so, for example, the first entry in the table does not actually constitute an apparent separation violation.

In the following data, time, t , is measured in seconds from 1 January 1970.

AAL2504 (DAL-LGA), lat =40.83, lon=-83.08, alt =319
LHN737 (BOS-IND), lat =40.83, lon=-83.02, alt =310
Incorrect separation violation: t =996674100, ind1=2, ind2=14.hdist =3.03 nm, vdist =900 ft

ABX1106 (ILN-BOS), lat =41.58, lon=-81.18, alt =277
COM477 (CVG-BGR), lat =41.62, lon=-81.27, alt =270
Incorrect separation violation: t =996694260, ind1=14, ind2=13.hdist =4.24 nm, vdist =700 ft

ASH758 (CMH-EWR), lat =41.23, lon=-81.75, alt =250
NWA71 (DCA-DTW), lat =41.25, lon=-81.80, alt =251
Incorrect separation violation: t =996691320, ind1=8, ind2=4.hdist =2.47 nm, vdist =100 ft

BLR6240 (CVG-BTV), lat =40.85, lon=-82.82, alt =270
BTA4233 (CLE-MCI), lat =40.78, lon=-82.78, alt =273
Incorrect separation violation: t =996699240, ind1=3, ind2=0.hdist =4.28 nm, vdist =300 ft

BTA3626 (CVG-EWR), lat =40.95, lon=-82.65, alt =279
UAL608 (ORD-DCA), lat =40.97, lon=-82.73, alt =270
Incorrect separation violation: t =996682740, ind1=4, ind2=0.hdist =3.91 nm, vdist =900 ft

BTA3863 (CLE-CLT), lat =41.43, lon=-81.78, alt =290
COA1682 (IND-EWR), lat =41.47, lon=-81.77, alt =290
Incorrect separation violation: t =996703020, ind1=0, ind2=11.hdist =2.14 nm, vdist =0 ft

BTA3863 (CLE-CLT), lat =41.43, lon=-81.78, alt =290
COA1894 (CLE-DCA), lat =41.43, lon=-81.78, alt =290
Incorrect separation violation: t =996703020, ind1=0, ind2=0.hdist =0.00 nm, vdist =0 ft

BTA3949 (CLE-MEM), lat =41.07, lon=-82.23, alt =278
UAL782 (ORD-BWI), lat =41.03, lon=-82.28, alt =270
Incorrect separation violation: t =996684000, ind1=1, ind2=2.hdist =3.02 nm, vdist =800 ft

BTA3974 (CLE-ISP), lat =41.53, lon=-81.75, alt =265
NWA1452 (DTW-ORF), lat =41.58, lon=-81.73, alt =258
Incorrect separation violation: t =996629220, ind1=1, ind2=1.hdist =3.09 nm, vdist =700 ft

BTA4130 (DAY-EWR), lat =41.25, lon=-81.83, alt =290
UAL1872 (ORD-PHL), lat =41.22, lon=-81.87, alt =290
Incorrect separation violation: t =996706680, ind1=10, ind2=4.hdist =2.50 nm, vdist =0 ft

BTA4132 (DAY-EWR), lat =41.32, lon=-81.15, alt =250
MES3637 (ABE-DTW), lat =41.27, lon=-81.10, alt =252
Incorrect separation violation: t =996664320, ind1=16, ind2=0.hdist =3.75 nm, vdist =200 ft

BTA4242 (CLE-MHT), lat =41.53, lon=-81.73, alt =287
N650TC (RKD-DAY), lat =41.50, lon=-81.68, alt =286
Incorrect separation violation: t =996630300, ind1=2, ind2=4.hdist =3.01 nm, vdist =100 ft

CGIWO (CYYZ-CMH), lat =41.40, lon=-82.20, alt =260
 UAL1852 (ORD-PHL), lat =41.33, lon=-82.20, alt =269
 Incorrect separation violation: t =996670860, ind1=3, ind2=2.hdist =4.00 nm, vdist =900 ft

COA1256 (CMH-EWR), lat =41.15, lon=-82.18, alt =257
 COM428 (CYYZ-CVG), lat =41.10, lon=-82.27, alt =260
 Incorrect separation violation: t =996688200, ind1=5, ind2=6.hdist =4.82 nm, vdist =300 ft

COA1415 (CLE-ORD), lat =41.50, lon=-81.92, alt =244
 USA238 (GRR-PIT), lat =41.52, lon=-81.93, alt =240
 Incorrect separation violation: t =996684060, ind1=2, ind2=4.hdist =1.25 nm, vdist =400 ft

COA1682 (IND-EWR), lat =41.47, lon=-81.77, alt =290
 COA1894 (CLE-DCA), lat =41.43, lon=-81.78, alt =290
 Incorrect separation violation: t =996703020, ind1=11, ind2=0.hdist =2.14 nm, vdist =0 ft

COA1900 (CLE-LGA), lat =41.55, lon=-82.03, alt =267
 UAL698 (ORD-LGA), lat =41.62, lon=-82.08, alt =275
 Incorrect separation violation: t =996666480, ind1=2, ind2=3.hdist =4.59 nm, vdist =800 ft

COA1904 (CLE-LGA), lat =41.62, lon=-81.23, alt =250
 EGF527 (DTW-LGA), lat =41.57, lon=-81.32, alt =255
 Incorrect separation violation: t =996697020, ind1=0, ind2=1.hdist =4.79 nm, vdist =500 ft

COA275 (CLE-LAX), lat =40.93, lon=-82.45, alt =288
 UAL366 (ORD-IAD), lat =40.93, lon=-82.48, alt =290
 Incorrect separation violation: t =996710400, ind1=1, ind2=2.hdist =1.51 nm, vdist =200 ft

COM599 (CVG-HPN), lat =40.70, lon=-82.98, alt =286
 DAL318 (CVG-JFK), lat =40.67, lon=-83.03, alt =290
 Incorrect separation violation: t =996693060, ind1=1, ind2=0.hdist =3.03 nm, vdist =400 ft

DAL43 (LFPG-CVG), lat =40.60, lon=-82.43, alt =289
 SWIFT64 (BLV-PSM), lat =40.58, lon=-82.38, alt =290
 Incorrect separation violation: t =996685380, ind1=1, ind2=0.hdist =2.49 nm, vdist =100 ft

EGF261 (ORD-PIT), lat =41.40, lon=-82.43, alt =245
 SYX2127 (CLE-MKE), lat =41.47, lon=-82.40, alt =254
 Incorrect separation violation: t =996695400, ind1=1, ind2=5.hdist =4.27 nm, vdist =900 ft

FDX3709 (IND-EWR), lat =40.67, lon=-83.10, alt =290
 N39TT (BKL-M33), lat =40.70, lon=-83.02, alt =281
 Incorrect separation violation: t =996703320, ind1=0, ind2=14.hdist =4.29 nm, vdist =900 ft

MEP82 (MCI-LGA), lat =41.33, lon=-82.28, alt =277
 TWA350 (STL-EWR), lat =41.35, lon=-82.18, alt =270
 Incorrect separation violation: t =996627660, ind1=2, ind2=10.hdist =4.61 nm, vdist =700 ft

N14RM (EYW-PTK), lat =40.85, lon=-82.10, alt =275
 USA30 (LAX-PHL), lat =40.83, lon=-82.15, alt =270
 Incorrect separation violation: t =996632400, ind1=2, ind2=5.hdist =2.48 nm, vdist =500 ft

N200MT (CYHU-LUK), lat =40.90, lon=-81.35, alt =286
 UAL1071 (MDT-ORD), lat =40.88, lon=-81.38, alt =280
 Incorrect separation violation: t =996704880, ind1=2, ind2=4.hdist =1.81 nm, vdist =600 ft

N501LS (CMH-TEB), lat =40.95, lon=-81.68, alt =256
 UAL749 (PHL-ORD), lat =40.92, lon=-81.70, alt =260
 Incorrect separation violation: t =996664440, ind1=6, ind2=6.hdist =2.14 nm, vdist =400 ft

N545S (MSP1-IAD), lat =40.87, lon=-81.65, alt =290
 USA269 (PIT-MKE), lat =40.82, lon=-81.68, alt =285

Incorrect separation violation: t=996704520, ind1=10, ind2=1.hdist =3.36 nm, vdist =500 ft

NWA1415 (ORF-DTW), lat =41.30, lon=-81.80, alt =243

USA953 (CLE-PHL), lat =41.33, lon=-81.88, alt =250

Incorrect separation violation: t=996665280, ind1=4, ind2=0.hdist =4.25 nm, vdist =700 ft

NWA1782 (DTW-PHL), lat =41.28, lon=-81.40, alt =265

NWA1783 (PHL-DTW), lat =41.30, lon=-81.43, alt =274

Incorrect separation violation: t=996707940, ind1=5, ind2=2.hdist =1.80 nm, vdist =900 ft

NWA571 (CLE-MSP), lat =41.40, lon=-81.93, alt =253

USA2332 (PHL-DTW), lat =41.37, lon=-81.97, alt =246

Incorrect separation violation: t=996666000, ind1=0, ind2=5.hdist =2.50 nm, vdist =700 ft

OPT316 (CGF-BED), lat =41.68, lon=-81.45, alt =268

UPS484 (PHL-DTW), lat =41.67, lon=-81.52, alt =268

Incorrect separation violation: t=996653280, ind1=1, ind2=2.hdist =3.15 nm, vdist =0 ft

SWA124 (CLE-BWI), lat =40.93, lon=-81.13, alt =284

UAL379 (PHL-ORD), lat =40.88, lon=-81.15, alt =280

Incorrect separation violation: t=996689940, ind1=7, ind2=3.hdist =3.09 nm, vdist =400 ft

SWA360 (CLE-MDW), lat =41.48, lon=-82.35, alt =295

USA420 (MKE-PIT), lat =41.45, lon=-82.35, alt =290

Incorrect separation violation: t=996682920, ind1=5, ind2=2.hdist =2.00 nm, vdist =500 ft

SYX2288 (MKE-IAD), lat =40.92, lon=-82.18, alt =250

USA2153 (PIT-MSP), lat =40.97, lon=-82.15, alt =244

Incorrect separation violation: t=996630000, ind1=3, ind2=8.hdist =3.36 nm, vdist =600 ft

SYX2417 (MKE-DCA), lat =40.93, lon=-82.18, alt =270

USA397 (PIT-SBN), lat =40.95, lon=-82.08, alt =272

Incorrect separation violation: t=996628620, ind1=3, ind2=9.hdist =4.64 nm, vdist =200 ft

36 collisions total

Origin= *, Destination= * Period = 0:00 to 24:00

References

- ¹ Steven M. Green, Karl Bilimoria, and Mark Ballin, “Distributed Air-Ground Traffic Management for En Route Flight Operations,” AIAA2000-4064, AIAA Guidance, Navigation, and Control Conference (August 2000).
- ² Steven M. Green and Robert Vivona, “Local Traffic Flow Management Concept for Constrained En Route Airspace Problem,” AIAA 2001-4115, American Institute of Aeronautics and Astronautics (2001).
- ³ Craig W. Reynolds, “Steering Behaviors For Autonomous Characters,” <http://www.red3d.com/cwr/steer/gdc99/index.html>.
- ⁴ Lee Berry, Bob Dory, et al., “An Analysis of the National Airspace Capacity,” Martin Marietta Energy Systems Report K/DSRD-1098 (September 1992).
- ⁵ Mark D. Ridgers, Richard H. Mogford, and Leslye S. Migford, “The Relationship of Sector Characteristics to Operational Errors,” DOT/FAA/AM-98-14 (May 1998).
- ⁶ Interviews with Cleveland center personnel Mr. Dan Wiita, ZOB48 controller, and Mr. Ted Piskur, Area 4 Supervisor for Cleveland center.
- ⁷ The Royal Aeronautical Society, “The Future of Aviation—Response to the DETR Consultative Document,” http://www.raes.org.uk/detr_response.pdf (April 2001).
- ⁸ Alvin L. McFarland, “A Conflict Probe to Provide Early Benefits for Airspace Users and Controllers,” The MITRE Corporation, <http://www.caasd.org/library/papers/uret/index.html>.
- ⁹ ir. Wim Post, Ian Wilson, “The PHARE experience with Conflict Detection and Resolution,” <http://www.eurocontrol.be/projects/eatchip/faa-euro/AP-group-meetings/AP6/tim6/PHARE/phare.htm>.
- ¹⁰ Ronald Azuma, Howard Neely III, Michael Daily, and Ryan Geiss, “Visualization Tools for Free Flight Air-Traffic Management,” IEEE Computer Graphics and Applications, September/October 2000 (2–6);
- ¹¹ “PHARE : Definition and use of Tubes,” DOC 96-70-18, Programme for Harmonised Air Traffic Management Research in Eurocontrol, Brussels, Belgium (15 July 1996).
- ¹² Paul Bourke, <http://astronomy.swin.edu.au/pbourke/geometry/2circle/>.