Project Report
ATC-219

Implementation of the 1992 Terminal Area-Local Analysis and Prediction System (T-LAPS)

S. H. Olson
C. D. McLain
S. Yao

22 August 1994

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

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

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

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
ATC-219		

4. Title and Subtitle	5. Report Date
Implementation of the 1992 Terminal area-Local Analysis and Prediction System (T-LAPS)	22 August 1994
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Stephen H. Olson, Cynthia D. McLain, and Shian-Shi Yao	ATC-219
9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Lincoln Laboratory, MIT P.O. Box 73 Lexington, MA 02173-9108	11. Contract or Grant No. DTFA01-91-Z-02036
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Department of Transportation Federal Aviation Administration Washington, DC 20591	Project Report
	14. Sponsoring Agency Code

15. Supplementary Notes

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. The work was sponsored by the Air Force under Contract F19628-90-C-0002.

16. Abstract

The Integrated Terminal Weather System (ITWS) development program was initiated by the Federal Aviation Administration (FAA) to produce a fully automated, integrated terminal weather information system to improve the safety, efficiency, and capacity of terminal area aviation operations. The ITWS will acquire data from FAA and National Weather Service (NWS) sensors as well as from aircraft in flight in the terminal area. The ITWS will provide air traffic personnel with products that are immediately usable without further meteorological interpretation. These products include current terminal area weather and short-term (0–30 minute) predictions of significant weather phenomena.

The Terminal area-Local Analysis and Prediction System (T-LAPS) is being evaluated as a possible provider of the Terminal Winds Product for the ITWS. T-LAPS is a direct descendant of the Local Analysis and Prediction System (LAPS) developed at the National Oceanic and Atmospheric Administration's (NOAA's) Forecast Systems Laboratory (FSL). T-LAPS takes meteorological data from a wide variety of data sources as input and provides a gridded, three-dimensional (3-D) analysis of the state of the local atmosphere in the terminal area as output. For the 1992 system, the output was a gridded 3-D analysis of the horizontal winds. This information is intended to be used by the Terminal Air Traffic Control Automation (TATCA) program to estimate the effects of winds on aircraft in the terminal area. The 1993 and 1994 T-LAPS systems will incorporate more sophisticated wind analysis algorithms.

The T-LAPS '92 demonstration at the Lincoln Laboratory Terminal Doppler Weather Radar (TDWR) FL-2C field site in Kissimmee, Florida, during August and September was quite successful. The primary area of coverage was a 120 km by 120 km box centered on the Orlando International Airport. The T-LAPS system was able to utilize radar information from both the TDWR testbed and the operational NEXRAD/WSR-88D radar in Melbourne, Florida. This report documents the implementation of the T-LAPS system that was run during the 1992 summer demonstration and discusses the design and some implementation details of the system.

17. Key Words		18. Distribution Statement
ITWS T-LAPS Terminal Winds demonstration	gridded analysis three-dimensional analysis real time	This document is available to the public through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	56	

FORM DOT F 1700.7 (8-72) Reproduction of completed page authorized

ABSTRACT

The Integrated Terminal Weather System (ITWS) development program was initiated by the Federal Aviation Administration (FAA) to produce a fully automated, integrated terminal weather information system to improve the safety, efficiency and capacity of terminal area aviation operations. The ITWS will acquire data from FAA and National Weather Service (NWS) sensors as well as from aircraft in flight in the terminal area. The ITWS will provide air traffic personnel with products that are immediately usable without further meteorological interpretation. These products include current terminal area weather and short–term (0–30 minute) predictions of significant weather phenomena.

The Terminal area–Local Analysis and Prediction System (T–LAPS) is being evaluated as a possible provider of the Terminal Winds Product for the ITWS. T–LAPS is a direct descendant of the Local Analysis and Prediction System (LAPS) developed at the National Oceanic and Atmospheric Administration's (NOAA's) Forecast Systems Laboratory (FSL). T–LAPS takes meteorological data from a wide variety of data sources as input and provides a gridded, three–dimensional (3–D) analysis of the state of the local atmosphere in the terminal area as output. For the 1992 system, the output was a gridded 3–D analysis of the horizontal winds. This information is intended to be used by the Terminal Air Traffic Control Automation (TATCA) program to estimate the effects of winds on aircraft in the terminal area. The 1993 and 1994 T–LAPS systems will incorporate more sophisticated wind analysis algorithms.

The T–LAPS '92 demonstration at the Lincoln Laboratory Terminal Doppler Weather Radar (TDWR) FL–2C field site in Kissimmee, Florida during August and September was quite successful. The primary area of coverage was a 120 km by 120 km box centered on the Orlando International Airport. The T–LAPS system was able to utilize radar information from both the TDWR testbed and the operational NEXRAD/WSR–88D radar in Melbourne, Florida. This report documents the implementation of the T–LAPS system that was run during the 1992 summer demonstration and discusses the design and some implementation details of the system.

TABLE OF CONTENTS

Section	Page
Abstract	iii
List of Illustrations	vii
List of Tables	vii
1. INTRODUCTION	1
2. T–LAPS '92 STRUCTURE	3
2.1. Data Ingest Module	3
2.2. Analysis Module	4
2.2.1. 10 km/2 km Cascade	4
2.2.2. Wind Package	6
2.2.3. Balance Package	7
2.3. Real–Time Driver/Scheduler	8
2.4. Displays	8
3. DATA INGEST	11
3.1. MAPS Data	11
3.2. Surface Station Data	13
3.3. ACARS Data	13
3.4. LLWAS Data	14
3.5. Radar Data	15
3.5.1. TDWR	17
3.5.2. NEXRAD	18
4. MAIN ANALYSIS PROGRAMS	21
4.1. Porting the FSL Software.	21
4.2. Specific Porting Changes	22
4.2.1. Pathnames	22
4.2.2. Logical vs. Integer Variables	23
4.2.3. VMS Time Representation	23
4.3. Wind Analysis Package	23
4.3.1. Processing of Radar Data	23
4.3.2. Barnes Interpolation	24
4.3.3. Miscellaneous Changes	25
4.4. Balance Package	26
5. REAL–TIME DRIVER	29
5.1. System Start Up	31
5.2. Interpolation of Wind Analysis Background	31
5.2.1. Spatial Interpolation	31
5.2.2. Temporal Interpolation	31

TABLE OF CONTENTS (CONTINUED)

Section **Page**

5.2.3. Height Adjustment 32
5.3. T–LAPS 10 km Processing 32
5.3.1. 10 km Data Ingest 34
5.3.2. T–LAPS 10 km Analysis 34
5.4. T–LAPS 2 km Processing 34
5.4.1. 2 km Data Ingest 34
5.4.2. T–LAPS 2 km Analysis and Display 36
5.5. Data Archive 36
5.6. Driver Communications with Independent Programs 36
5.7. Missing Data Conditions 36
5.8. System Shut Down 37

6. DISPLAY 39

6.1. Design Considerations 39
6.2. Display Product Server 39
6.2.1. Initialization 39
6.2.2. Data Ingest 39
6.2.3. Processing 41
6.2.4. Product Serving 42
6.2.5. Termination 42
6.3. Product Display 42
6.3.1. Initialization 42
6.3.2. Data Ingest 44
6.3.3. Display Processing 44
6.3.4. Display Termination 44

7. ARCHIVING 45

7.1. Storage of T–LAPS Data at FL–2C 45
7.2. Archiving to Tape at FL–2C 47
7.3. Archiving via File Transfer 47
7.3.1. From External Agencies 47
7.3.2. From the T–LAPS Real–Time System Disks 47
7.3.3. From Real–Time Data Feeds 48
7.3.4. Archiving by Others 48
7.4. Organization of the T–LAPS Database 48

8. CONCLUSION 51

GLOSSARY 53

REFERENCES 55

LIST OF ILLUSTRATIONS

Figure No.		Page
1.	T–LAPS block diagram.	3
2.	LAPS wind analysis.	5
3.	T–LAPS wind analysis 10 km/2 km cascade.	5
4.	Nested T–LAPS domains.	6
5.	Sample display showing wind arrows and gray–scale reflectivity.	9
6.	Example of MAPS availability times.	12
7.	Example of MAPS valid times.	12
8.	Surface station locations.	14
9.	Orlando runways and LLWAS stations.	15
10.	T–LAPS domain showing radars.	18
11.	T–LAPS processes and hosts.	30
12.	MAPS grid points.	32
13.	T–LAPS 10 km processing.	33
14.	T–LAPS 2 km processing.	35
15.	Format of T–LAPS display used at FL–2C.	40
16.	Display product server processing.	41
17.	Real–time product display.	43
18.	Relationship between physical disks and working space.	46
19.	Data flow into the T–LAPS database.	49

LIST OF TABLES

Table No.		Page
1.	Approximate T–LAPS Space Requirements for One Hour of Operations	46

1. INTRODUCTION

The Integrated Terminal Weather System (ITWS) development program was initiated by the Federal Aviation Administration (FAA), [Sankey and Hansen, 1993] to produce a fully automated, integrated terminal weather information system to improve the safety, efficiency and capacity of terminal area aviation operations. The ITWS will acquire data from FAA and National Weather Service (NWS) sensors as well as from aircraft in flight in the terminal area. The ITWS will provide air traffic personnel with products that are immediately usable without further meteorological interpretation. These products include current terminal area weather and short–term (0–30 minute) predictions of significant weather phenomena.

The Terminal area–Local Analysis and Prediction System (T–LAPS) is being evaluated as a possible provider of the Terminal Winds Product for the FAA's ITWS [Evans, 1991; Ducot, 1993]. The Terminal Winds product is to provide a gridded three–dimensional analysis of the winds in the terminal area. The purpose of this report is to document the implementation of the T–LAPS system that was run during the Orlando 1992 summer demonstration. This report will discuss the design and some implementation details of the system. A detailed discussion of algorithm design or algorithm performance during the demonstration is beyond the scope of this report.

T–LAPS takes as input meteorological data from a wide variety of data sources, including current sensor data and background information from larger–scale analyses. The output of T–LAPS is a gridded, three–dimensional (3–D) analysis of the state of the local atmosphere. For the 1992 system, the output was a gridded 3–D analysis of the horizontal winds. This information in turn is intended to be used by the Terminal Air Traffic Control Automation (TATCA) [Andrews and Welch, 1989] program in order to estimate the effects of winds on aircraft in the terminal area.

T–LAPS is a direct descendant of the Local Analysis and Prediction System (LAPS) [McGinley, *et al.*, 1991], which was developed at the National Oceanic and Atmospheric Administration's (NOAA's) Forecast Systems Laboratory (FSL). LAPS is a large and complex system that produces a fairly complete analysis of the state of the local atmosphere. T–LAPS '92 was built around the subset of LAPS that is relevant for computing gridded 3–D horizontal winds. Additions were made to the LAPS core system in order to adapt it to the needs of the terminal environment. The most important changes were adapting the system to work on the small scale of the terminal environment (a 120 km by 120 km domain versus LAPS' original 600 km by 600 km domain) and upgrading the system so that it could make use of radar Doppler measurements from multiple radars. T–LAPS is an ongoing project, but this report will only discuss only the 1992 version of the system. In the immediate future (1993, 1994), the system will incorporate more sophisticated wind analysis algorithms.

The centerpiece of the 1992 T–LAPS effort was a demonstration of the system at Lincoln Laboratory's Terminal Doppler Weather Radar (TDWR) testbed [Merritt, *et al.*, 1989] FL–2C field site in Kissimmee, Florida during August and September. The primary area of coverage was a 120 km by 120 km box centered on the Orlando International Airport in Orlando, Florida. Notably, the T–LAPS system was able to utilize radar information from both the TDWR prototype radar in Kissimmee and the operational NEXRAD/WSR–88D radar [Crum and Alberty, 1993] located in Melbourne, Florida. Fielding the system involved porting the initial FSL software, modifying the analysis software for the terminal environment, arranging for the ingest of multiple data sources, constructing a display, arranging for the archiving of data, and making the system work reliably.

The T–LAPS '92 demonstration was quite successful. The system was up and running, collecting and recording data within a few days of the availability of the last significant data source in August. The system ran reliably until the demonstration was terminated due to lack of significant weather in late September. The T–LAPS '92 demonstration marked the first demonstration of a prototype ITWS product using a wide variety of data sources. The resulting dataset became the basis for further algorithm development.

The report is organized as follows. Section 2 is an extended introduction that discusses the functionality of each of T–LAPS's major modules, with emphasis on showing how the whole system fits together. The remainder of the report discusses implementation issues, showing how the original FSL code was turned into a deployable system. Section 3 focuses on data acquisition. Section 4 provides details about the process of porting the FSL LAPS code to the Lincoln Laboratory computer environment. Section 5 discusses the real–time scheduler/driver. Section 6 covers a display system for presenting the results of the system. Section 7 discusses the archiving of results for later playback and analysis. Section 8 is a short concluding section.

2. T–LAPS '92 STRUCTURE

The task of the T–LAPS '92 system was to acquire data from various data sources, analyze the data to obtain a representation of the gridded 3–D horizontal winds, display the results, and save the results for further analysis. Accordingly, there were four major modules of the system: a data ingest module, a wind analysis module, a display module and a data archive module. The fifth major module was a driver/scheduler to coordinate the activities of the other four modules. The archive module was not really a separate, identifiable piece of software; the code that does exist is part of the driver. A simplified block diagram of the system is shown in Figure 1.

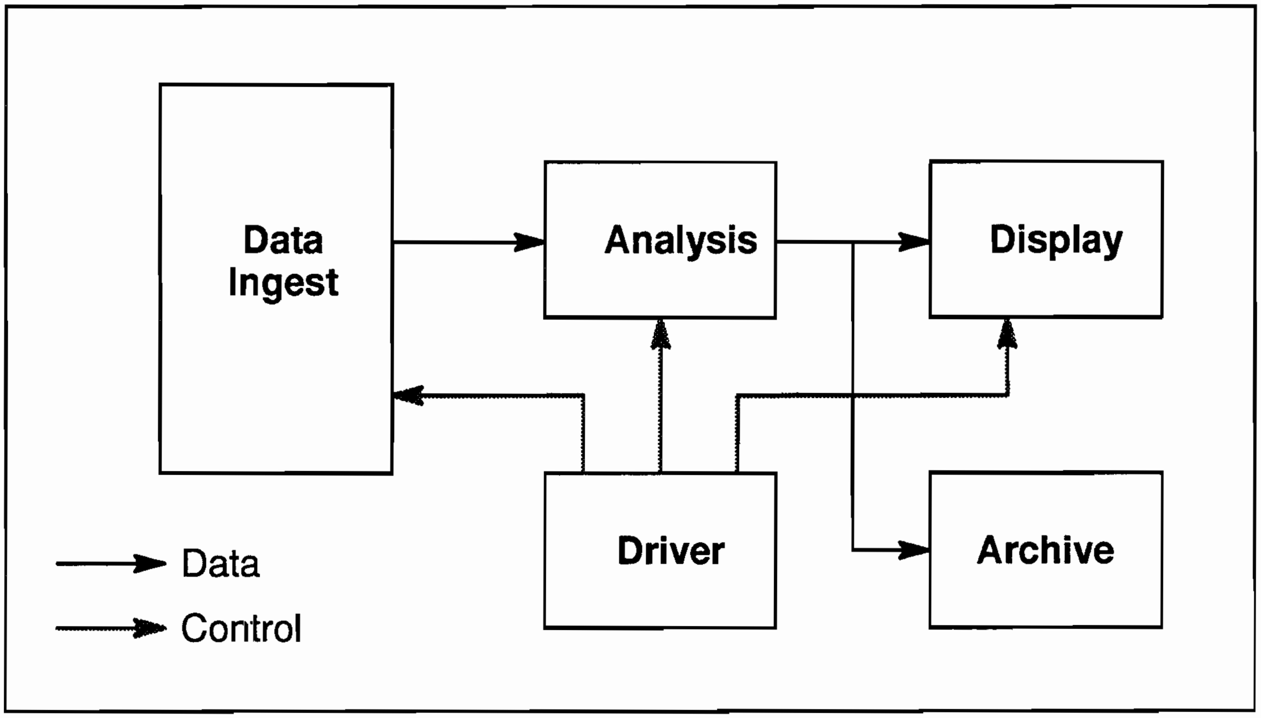

Figure 1. T–LAPS block diagram.

2.1. Data Ingest Module

The purpose of the Data Ingest Module was to access data from sources external to the T–LAPS system. This was a difficult task because of the number and variety of the different data sources. The Data Ingest Module was a collection of independent submodules that each handled a different data source. The data sources that were already accessible at the FL–2C site were Doppler radar data and Low Level Windshear Alert System (LLWAS) data. The radar data were resampled from their original polar coordinate system into a Cartesian coordinate system that T–LAPS could use. T–LAPS had to acquire other sets of data from outside organizations via telephone lines. Surface aviation observations (SAOs) were acquired from a NASA facility at Cape Canaveral. Automated pilot reports generated by the Aircraft Communications Addressing and Reporting System/Meteorological Data Collection and Reporting System (ACARS/MDCRS) were acquired from Aeronautical Radio, Inc.

(ARINC). Data from a national–scale gridded analysis system called the Mesoscale Analysis and Prediction System (MAPS) were acquired from FSL. At times, the acquisition of data from outside organizations was adversely affected by circumstances beyond the control of the T–LAPS system. A great deal of the complexity of the data acquisition submodules was devoted to robustly adapting to problems with the external data sources.

2.2. Analysis Module

The task of the analysis module was to assemble all the data sources into a single, gridded 3–D representation of the horizontal winds. To perform this task, the LAPS system uses two different sub–modules. The "wind package" interpolates the different data sources to produce the initial gridded winds, while the "balance package" uses the physics of the wind/pressure relationship to refine the initial gridded winds. The analysis module in the deployed T–LAPS '92 system consisted of three distinct components: the balance package and two separate versions of the wind package.

2.2.1. 10 km/2 km Cascade

In order to supply timely, high–resolution wind information, T–LAPS used a faster update rate and a higher–resolution analysis grid than does the original LAPS system. In order to take advantage of the high update rate and high–resolution data provided by Doppler weather radar, T–LAPS used an update rate of five minutes and a grid resolution of 2 km. This presented a problem because of the great disparity in both grid resolution and update rate between the MAPS data used as background and the wind analysis. To solve this problem, two levels of wind analysis were used. A first–level analysis was run in a manner very similar to the original LAPS. This first–level analysis, in turn, provided the background for a high–resolution, fast update second–level wind analysis. The two–levels of analysis idea is called the "cascade of scales."

LAPS takes the 60 km MAPS analysis (each MAPS grid point represents roughly a 60 km by 60 km area) and uses it as the background for its own 10 km analysis (each LAPS grid point represents roughly a 10 km by 10 km area). There is one version each of the wind package and the balance package, both using a 61–point by 61–point 10 km grid. The system has 21 vertical levels spaced 50 millibars apart. The vertical domain reaches from 1100 millibars (below sea level) to 100 millibars. The system is run once an hour. The structure of the LAPS wind analysis system is shown in Figure 2.

T–LAPS uses two scales of analysis. The internal structure of the T–LAPS Analysis Module is shown in Figure 3. In T–LAPS, the two blocks in the figure labeled **Winds_10** and **Balance_10** essentially recreated the manner in which LAPS works. Together these two packages are called the "10 km analysis." The 10 km wind package and the balance package were run every 30 minutes. The output of the balance package was then used as the background to the 2 km wind package. The 2 km wind package – **Winds_2** in the figure – was run every five minutes using the most recent 10 km result as its background. A five–minute update rate was chosen because it is approximately the same amount of time as a TDWR volume scan.

It would have been possible to jump directly from the 60 km MAPS to a 2 km wind analysis, but the disparities in scale would have been large. The update rate and data densities of both surface observations and ACARS are effectively utilized by an intermediate–scale analysis. The introduction of a second scale of analysis allowed the 10 km analysis to supply a higher quality background to the 2 km analysis than was possible by using MAPS directly.

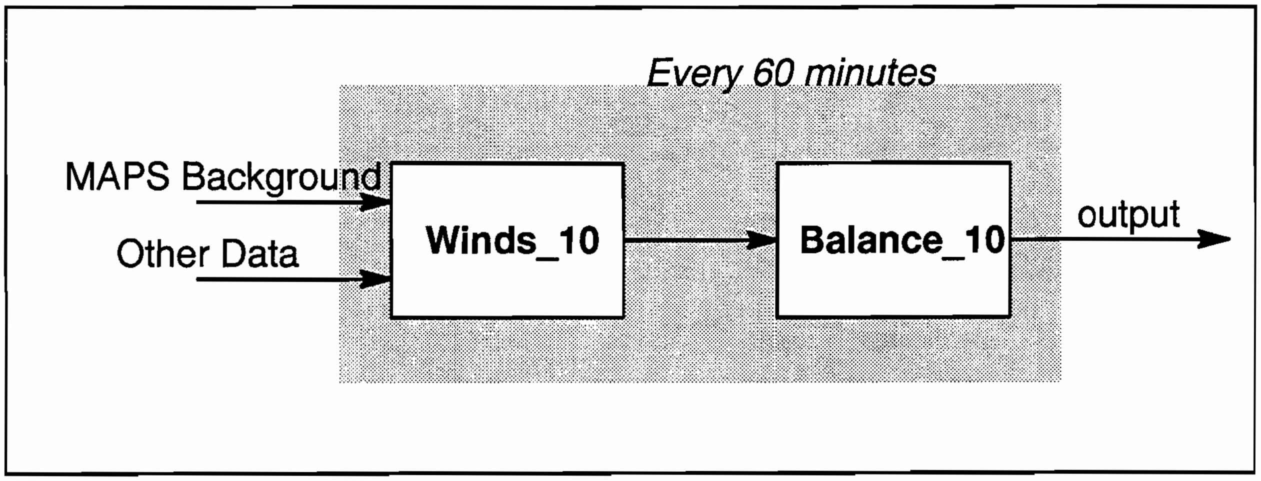

Figure 2. LAPS wind analysis.

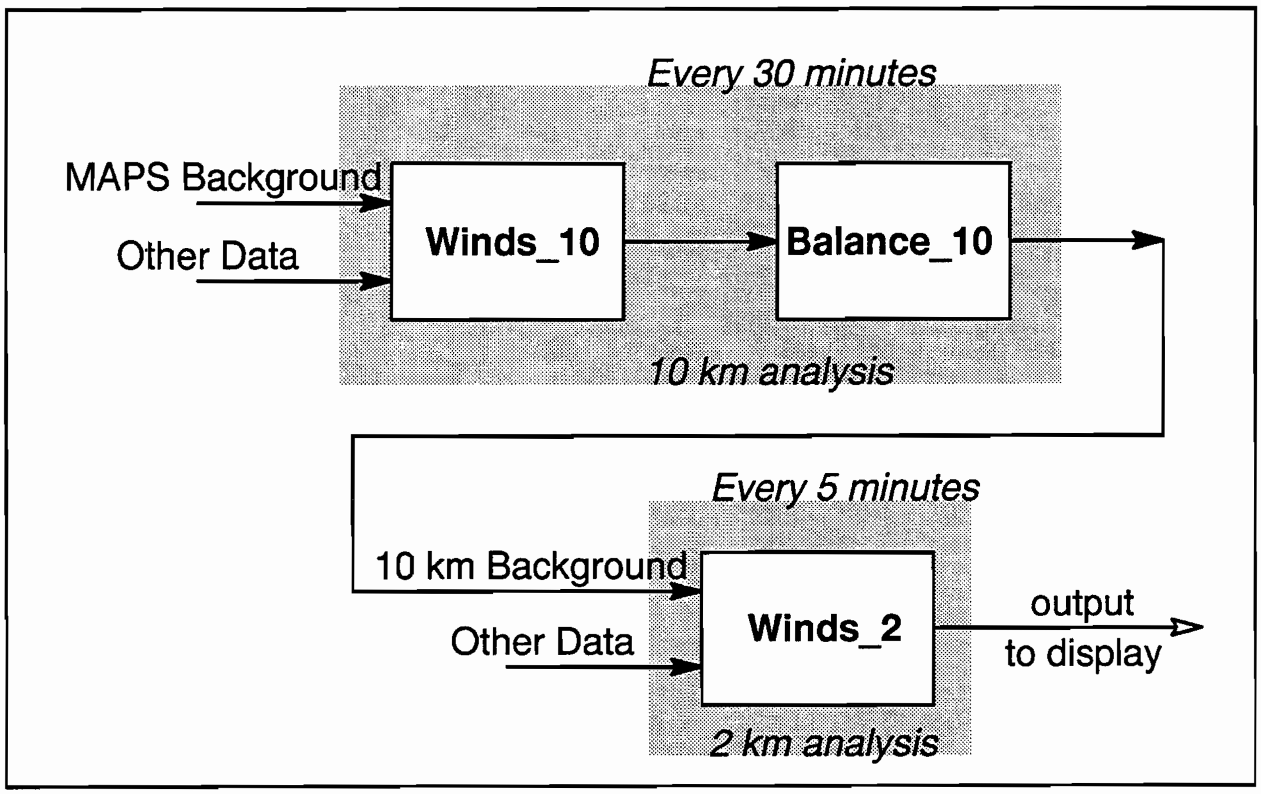

Figure 3. T–LAPS wind analysis 10 km/2 km cascade.

In T–LAPS '92 the 10 km analysis exists solely to provide a high–quality background for the 2 km analysis. Therefore, the size of the 10 km domain was only slightly larger than the size of the 2 km domain. For the 10 km analysis, a grid size of 19 by 19 was used, giving a total domain size of 180 km by 180 km (measuring between the nominal grid points, not the outer edges of the grid boxes). Because of the needs of the balance package, the full 21 vertical levels used in LAPS are also used in the 10 km analysis. The 2 km analysis used a 61 by 61 grid, giving a total domain size of 120 km by 120 km. To save file space and processing time, the 2 km analysis used only 13 vertical levels. Both domains were centered on the Orlando airport, so the 2 km domain nested inside the 10 km domain. Figure 4 shows the two domains superposed on a map of central Florida. The eastern Florida coast shown in the figure is the Cape Canaveral area, while the western Florida coast shown in the figure is the Tampa/St. Petersburg area.

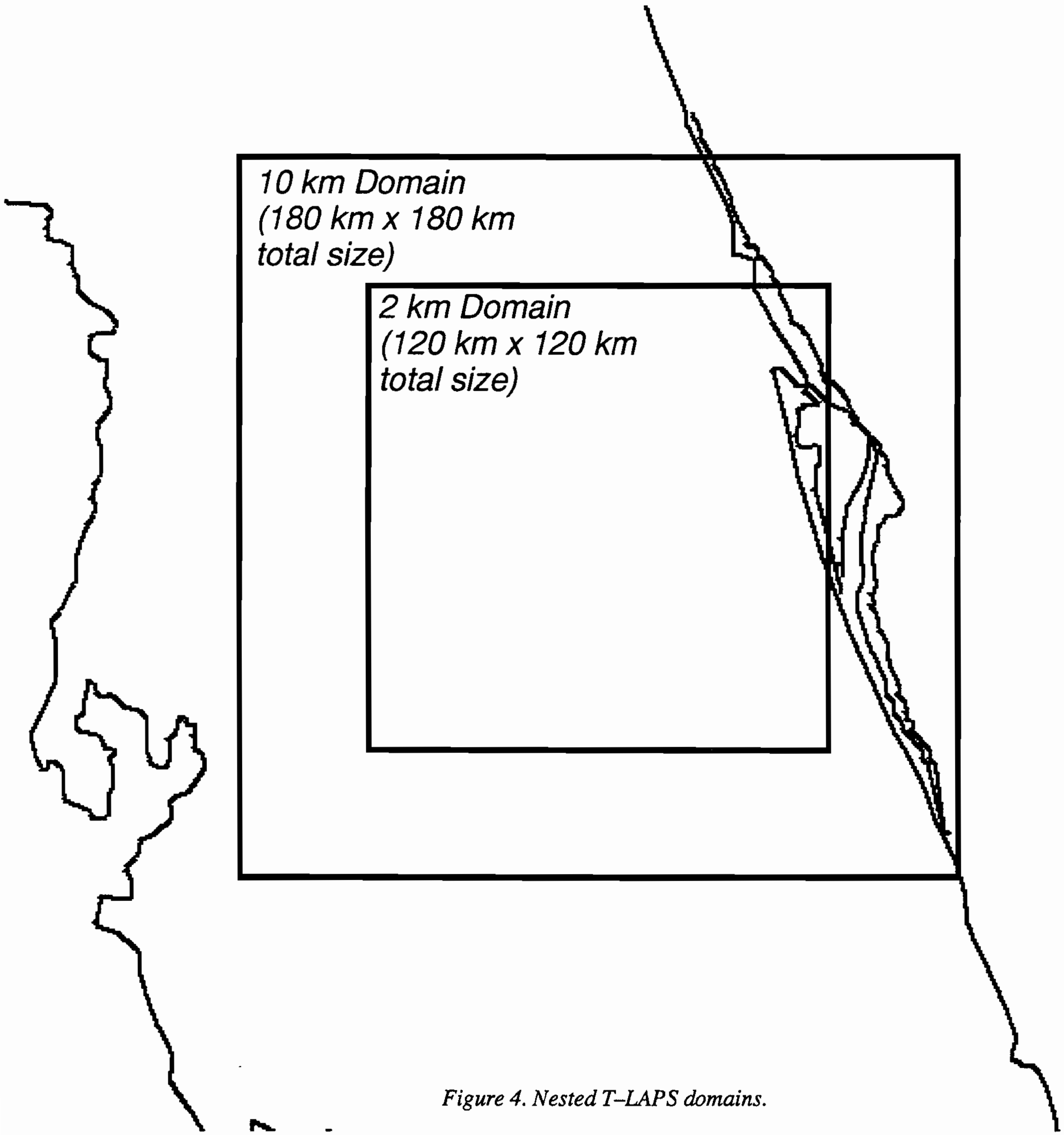

Figure 4. Nested T–LAPS domains.

2.2.2. Wind Package

The wind package constructed a 3–D grid of the horizontal wind field by interpolating together the different sources of wind information [Albers, 1992]. The underlying engine of the wind package is based on Barnes interpolation [Barnes, 1964]. Except for radar data, the number of observations from the various data sources is very small compared to the number of grid points. Barnes interpolation uses an exponential weighting scheme to obtain values for the empty grid points between the observations. The interpolation has a smoothing effect on grid points that do have observations. Two separate Barnes interpolation passes are used. In the first interpolation pass, all the vector (2–D horizontal) wind observations are interpolated to form a first–pass wind field. This includes data from surface observations, LLWAS, and ACARS. Next, the radial Doppler velocities are turned into synthetic vector observations. An azimuthal component for each radar observation is estimated by looking at the first–pass wind field. The estimated azimuthal component is combined with the radial radar velocity observation to produce the synthetic vector observation. In a second pass, all the original vector observations and the synthetic vector radar observations are interpolated to give the final result.

An important part of the wind analysis algorithm is the use of a background field. In the Barnes interpolation passes described above, the observations are not directly interpolated together. The deviation of each observation from the background field is computed and the deviations are interpolated together. The resulting field is then added to the background to produce the final result. When there are no observations, the background field becomes the final result. Additionally, the background is used for data quality control by disregarding observations that deviate too much from the background. MAPS data is used for the 10 km background, while the output of the 10 km analysis is used as the background for the 2 km wind analysis.

In the spring of '92 Lincoln Laboratory and FSL jointly designed, and FSL implemented, an upgrade to the wind package so that it utilizes Doppler information from multiple radars [Cole, *et al.*, 1993]. A detailed explanation of the "multi–single Doppler" algorithm is beyond the scope of this report, but it basically works as follows. If a grid point has two radar observations, then the first radar observation is handled essentially as before by comparing it against the first pass wind field and deriving a synthetic 2–D observation. The second observation is then handled by comparing it against the synthetic observation derived from the first observation. The algorithm is asymmetric in the manner in which it treats the two radars; the second radar is more important. If the radar beams are orthogonal to each other, then the method is essentially the same as standard dual Doppler. If the radar beams are parallel to each other, then the second radar observation overwrites the first radar observation. Thus, the most trusted radar near the airport – FL–2C – was used as the second radar, while the Melbourne NEXRAD/WSR–88D was used as the first radar.

2.2.3. Balance Package

The balance package is applied to the output of the wind package. The balance package also uses pressure information from MAPS and vertical wind information from various other LAPS modules. The balance package modified the pressure field and the wind field by using them as mutual constraints upon one another. The package used an iterative adjustment scheme to make the pressure and wind fields be consistent with each other in accordance with meteorological formulas that describe the wind/pressure relationship. There was no attempt by Lincoln Laboratory to modify the balance package algorithm.

The balance package counts on having no air escaping out of its top and bottom boundaries, so the vertical domain of the balance package needs to extend from the bottom of the atmosphere (the ground) to nearly the top of the atmosphere. The vertical domain of the wind package must match that of the balance package so the wind package must also use a vertical domain that extends almost to the top of the atmosphere.

2.3. Real–Time Driver/Scheduler

The driver is the module which controls the operation of the T–LAPS system. It is the timekeeper of the system and it schedules and coordinates the operations of the other modules. The timing requirements of the analysis module were relatively simple and the execution times of the individual analysis programs were relatively stable. The data ingest programs, however, had somewhat more complex timing requirements and much less predictable execution times. The design of the driver was simplified by exploiting the fact that the driver program and all the components of the analysis module were run on a single machine. This is not a very good model for future T–LAPS systems, but it was useful in '92 for reducing the level of effort needed to construct a viable system. The sophistication that is in the driver is devoted to flexibly adapting to missing or late data sources.

The communication between T–LAPS modules is primarily file–based, an arrangement that was inherited from the LAPS software. The driver provides the "glue" between the various modules by copying data files between the working directories used by the different modules and sub–modules. As a part of this functionality, the driver also arranges for important data files to be copied into special archive directories.

2.4. Displays

The displays were used to depict the results of the T–LAPS analysis. The real–time displays depicted the horizontal winds over the area of the T–LAPS domain in central Florida. Radar reflectivity and radial velocity information were also shown on these displays. The quantity of information depicted depended on the location of the display. Three displays were used during the demonstration period.

The primary display was located at the FL–2C site. At that installation, both the horizontal winds and the resampled radar radial velocity data were depicted for the surface to approximately 18,000 feet above ground level. The wind fields were stratified into 13 levels by the analysis programs. This display was used for monitoring T–LAPS performance and for demonstrations.

Secondary displays were located at Lincoln Laboratory in Lexington, Massachusetts and at the National Weather Service (NWS) office in Melbourne, Florida. Due to bandwidth constraints, only 5 levels of wind data were available for display at these locations. The Lexington display was used for monitoring both the weather situation and the performance of T–LAPS during real time. The NWS Melbourne display was used for demonstration purposes. For the secondary displays, only the surface level of radar radial velocity information was depicted. For all displays, only the surface level of radar reflectivity information was shown. Analysis results for each time interval were transmitted to the Lexington and Melbourne displays via dedicated telecommunications links.

Figure 5 illustrates the T–LAPS display. The wind speed and direction at each grid point are represented by an arrow. The reference arrow in the upper right corner provides a scale for the arrows. Radar reflectivity is shown using a simple gray–scale representation. The display shows a sea breeze front moving in from the east coast.

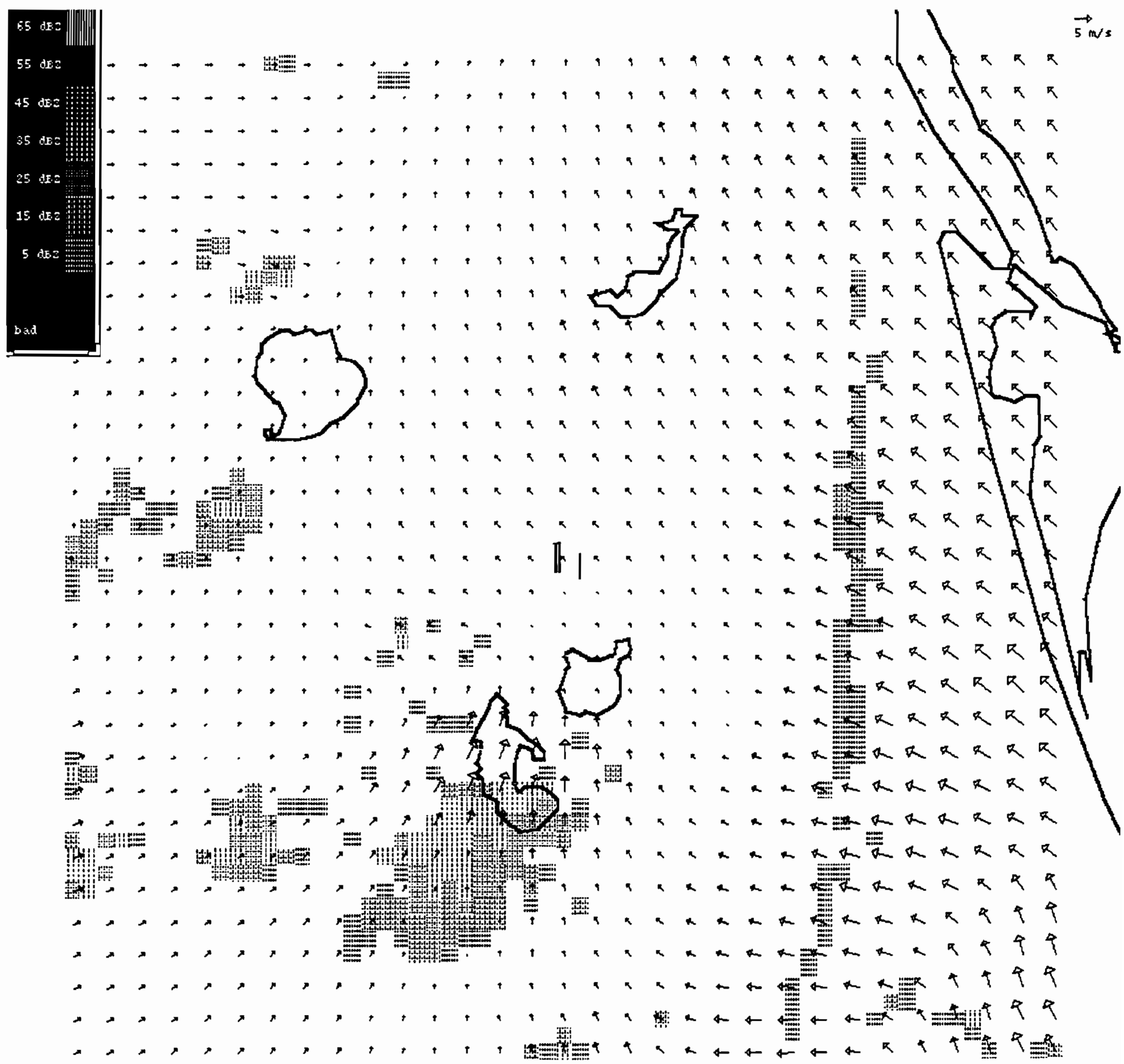

Figure 5. Sample display showing wind arrows and gray–scale reflectivity.

3. DATA INGEST

T–LAPS used five different data sources during the '92 demonstration. MAPS data, ACARS data, and surface station data were brought to the FL–2C site via telephone lines and then further processed into the formats used the by analysis programs. Radar data and LLWAS data were already available at site and were ingested using standard Lincoln software.

3.1. MAPS Data

The Mesoscale Analysis and Prediction System (MAPS), also developed at FSL, is a national–scale meteorological data analysis system [Benjamin, *et al.*, 1991; Schlatter, 1991]. MAPS is a prototype for the future Aviation Gridded Forecast System (AGFS) [Sherretz, 1991; Kraus, 1993], which will provide the background data for an operational ITWS. The domain of MAPS is the entire continental United States. It operates on an 81 by 62 grid with 25 vertical layers. Each grid point represents a nominal 60 km by 60 km area, although the actual area of the grid points varies somewhat due to the polar stereographic coordinate system used by MAPS.

MAPS operates on a three–hour time cycle. MAPS is keyed to Universal Time, so results are produced for the times 0:00Z, 3:00Z, 6:00Z, etc. It generates both a current analysis and three– and six–hour forecasts. The timeliness of MAPS generation is an important issue. The MAPS computation is delayed by FSL about 90 minutes in order to allow for all the input data to come in. Once the calculation is started it takes about 30 minutes to generate the current analysis and about 30 minutes more for each of the three–hour forecast and six–hour forecast. Thus, the MAPS analysis is not ready until about two hours after the effective time, while the three–hour forecast is ready about 30 minutes before the effective time. Figure 6 illustrates this process for the example cycle times of 12Z and 15Z

MAPS is available in three–hour increments, but the part of T–LAPS that uses MAPS data operates on a 30 minute cycle. To get around this difficulty, a synthetic MAPS data set is constructed at the desired time by interpolating between two MAPS data sets. The MAPS analysis arrives well after it is first needed, so the 3 hour and 6 hour forecasts are used. Thus to construct a synthetic MAPS analysis for the time 16:00Z, the 12:00Z 3 hour forecast and the 12:00Z 6 hour forecast would be interpolated. The 15:00Z current analysis and the 15:00Z 3 hour forecast would cover the same time period and would also suffice to construct a synthetic 16:00Z dataset, but those datasets would be not available in time for T–LAPS to use them in real–time. Figure 7 illustrates this choice; MAPS data from either the 12Z or 15Z cycle could be used to interpolate to 16Z, but Figure 6 shows that the 15Z data is not available until later.

MAPS data was obtained from FSL in Boulder, Colorado via standard telephone lines. In its original form, MAPS has a different product set and a different vertical coordinate system than does LAPS or T–LAPS. The FSL LAPS group performed the necessary translations and made the results available for dial–up transfer. MAPS subsets (6 by 6 horizontal grid points by 21 vertical layers) that covered the central Florida region were downloaded. FSL extracted the central Florida subset from full MAPS and encoded the data in LAPS format. There were two dial–ups every three hours: once to get the three–hour forecast file and once to get the six–hour forecast file. Two separate dial–ups were used because the three–hour forecast file was often needed before the six–hour forecast file was available. The actual time the files were ready was somewhat variable, depending on the state of the FSL computers. The dial–up software had to be configured to retry multiple times in case the files were not ready on the first attempt.

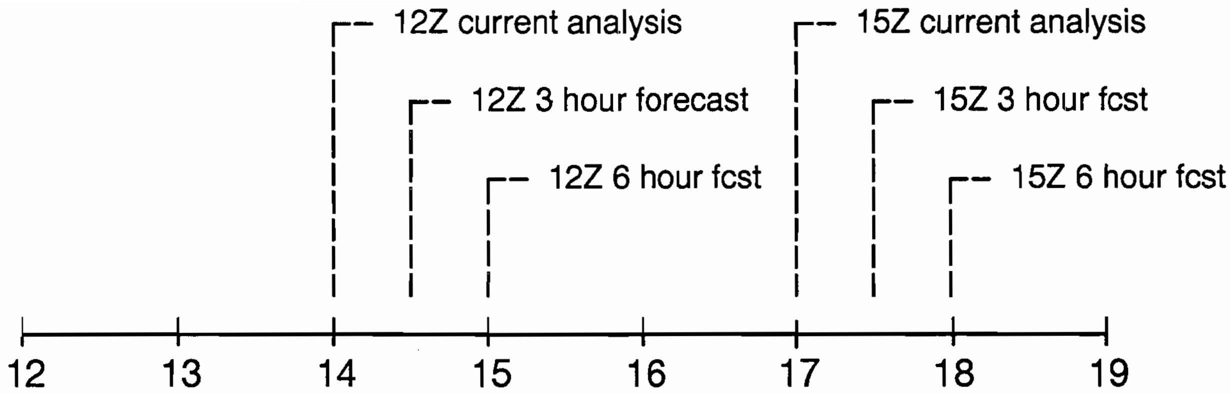

Figure 6. Example of MAPS availability times.

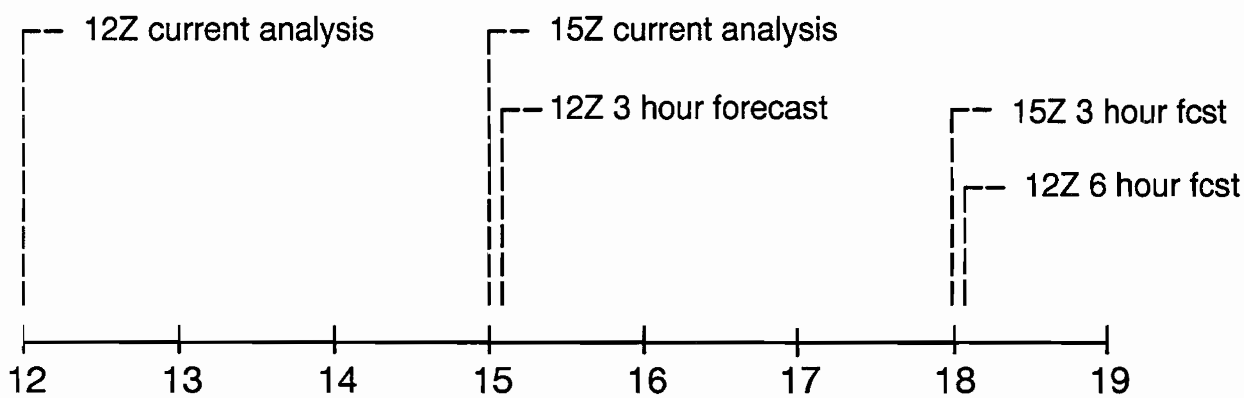

Figure 7. Example of MAPS valid times.

The communications protocol used was Kermit. The files were about 200k bytes in size and could be downloaded in about 2 minutes each. These files were formatted in a space–inefficient manner, and so the data compression built into the Kermit protocol was able to effectively double the transfer rate. Kermit worked very well. There were almost no problems with dropped connections or garbled data. There were problems, however, when the requested MAPS file was not available at FSL when the Kermit connection was established. In that situation, Kermit would abort without going through its normal logout procedure. This in turn, would "freeze" the software at the FSL end for about two hours until the FSL software automatically reinitialized itself. The problem was finally solved by making a small modification to the Kermit software so that it would go through its normal logout procedure even when a requested file was not found. After this problem was solved, the only times that MAPS data was not successfully downloaded were when FSL did not generate our MAPS files. This occurred only when the FSL computers were down (e.g., power failure at FSL or monthly maintenance) or when FSL was extremely late in generating the MAPS files.

3.2. Surface Station Data

Surface station data were obtained by telephone from a NASA facility at Cape Canaveral. The NASA facility has a data feed that brought them surface station data for the entire continental United States. For T–LAPS use, NASA pulled out a central Florida subset of the data. There were 19 surface stations in or near the T–LAPS domain. Most of the stations were SAOs, which is a system of manual observations with reports roughly every hour. Two of the stations were Automated Weather Observing Stations (AWOS) [NOAA–NWS, 1979], which report results about every 20 minutes.

In Figure 8, the solid dots show the locations of the surface stations. Note that some of the stations were actually outside of the T–LAPS domain. Station data from stations outside the domain were incorporated into the wind analysis by adjusting the station location information. The hollow diamonds show the adjusted locations that were supplied to the wind analysis in place of the real locations outside the domain. The locations of four stations in the Tampa Bay area to the west and one station to the north were adjusted this way.

NASA was dialed every 15 minutes to get the latest data; however, most of the surface stations reported results only once an hour. The data came over the line as simple text which was captured in a log file. Utilities were written to reformat it into the format required by the wind analysis program. The connection speed was only 2400 baud, but the size of the data was quite small – a few thousand characters of text – so transfer time was only one or two minutes.

Occasionally there were problems getting through because of third parties accessing the same modem at NASA. Because of internal constraints, NASA was unable to install an extra modem. To deal with this problem, the modem software was designed to simply wait a few minutes and re–dial. This generally worked well, but sometimes the data would not be available in time for the next T–LAPS cycle.

3.3. ACARS Data

The ACARS is a system for downlinking reports of meteorological information such as wind speed and direction from commercial aircraft. What was actually used was a subset of the ACARS data known as MDCRS [Dey, *et al.*, 1991; Martin, *et al.*, 1993]. A problem with ACARS data is that there is built–in latency in the system. The ACARS–equipped aircraft take measurements every 7.5 minutes, but the reports are bundled into groups of six and transmitted every 45 minutes. This means that the latency of any given ACARS report could be anywhere from 0 to 38 minutes. During the period August 17 to August 31, United Airlines had its ACARS–equipped planes provide reports every 2000 feet of elevation while either ascending from or descending to the Orlando Airport. This practice had the double benefit of providing vertical soundings near the airport and reducing data latency.

ACARS data was obtained from the commercial service provider, ARINC. It was a telephone connection using a Trellis modem and X.25 protocol software. A call was placed every 15 minutes to retrieve data. A data window of 90 minutes was specified, meaning that ACARS reports up to 90 minutes old were downloaded. We were able to re–use software that already existed at Lincoln to

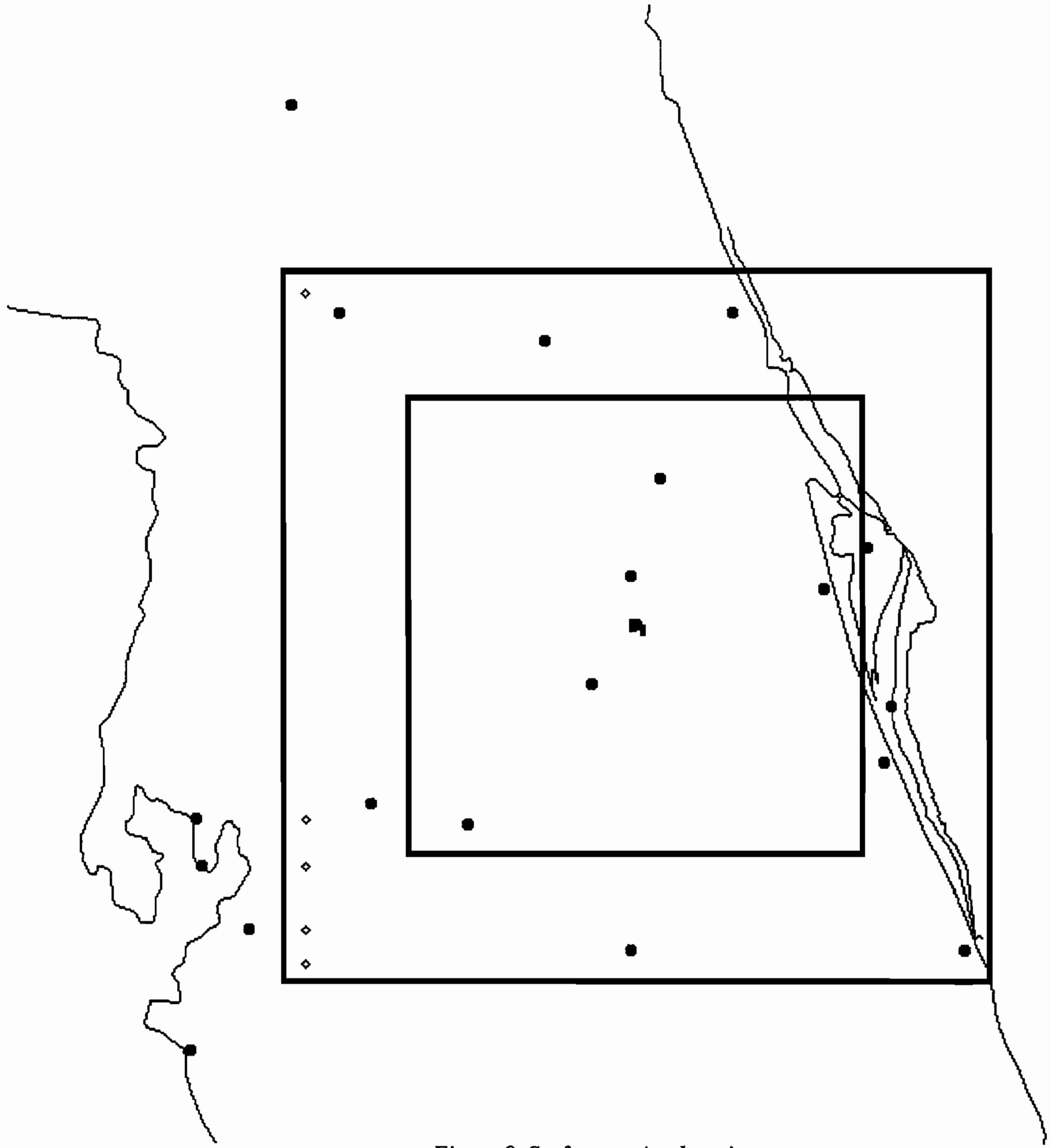
Figure 8. Surface station locations.

control the modem and request our data. The data came in the Binary Universal Form for the Representation of Meteorological Data (BUFR) format, so we had to write a utility to translate BUFR to the simple text–based format that T–LAPS used for pilot reports. The ACARS ingest was quite reliable.

3.4. LLWAS Data

The LLWAS [Wilson and Gramzow, 1991] is a network of ground–based anemometers that are clustered around the runways of an airport. From the perspective of T–LAPS, LLWAS data comes at a very high update rate (10 seconds) and is quite dense spatially. Figure 9 shows how the LLWAS stations are arranged around the Orlando Airport runways.

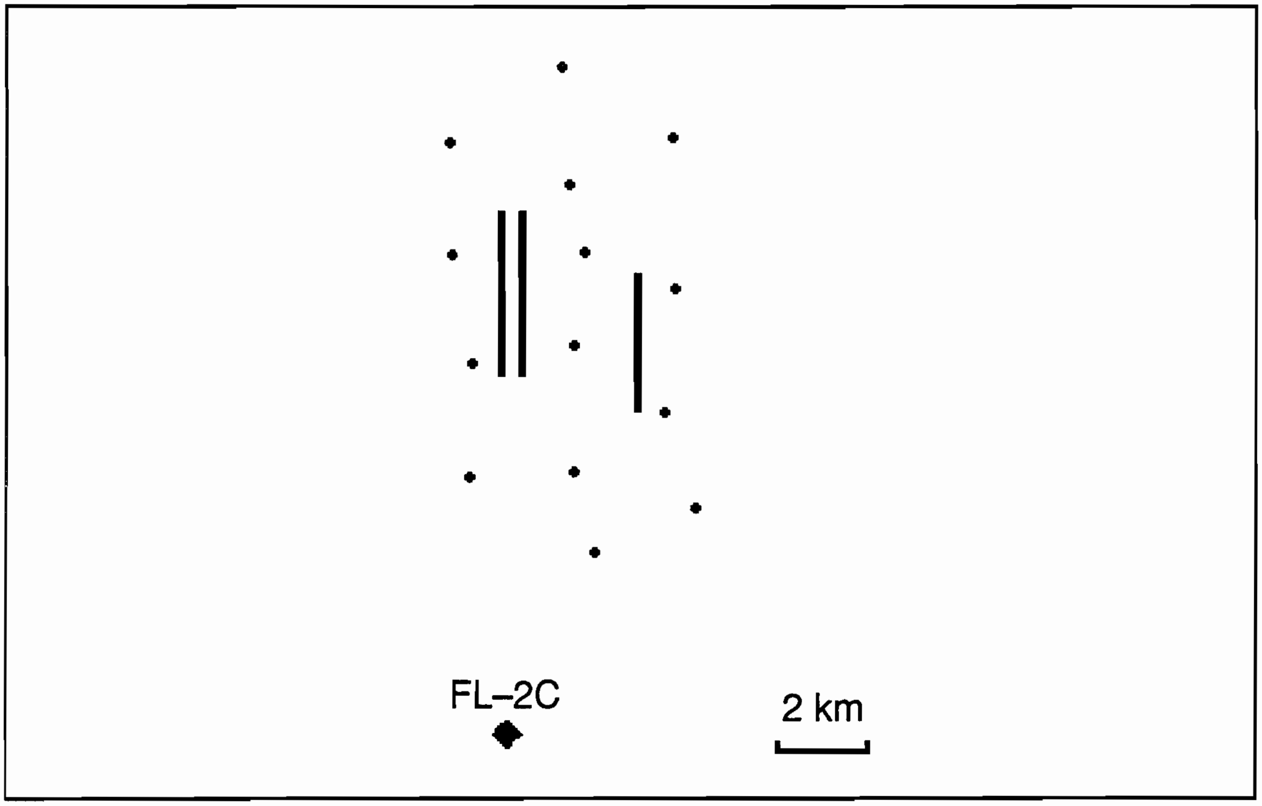

Figure 9. Orlando runways and LLWAS stations.

LLWAS data were ingested using facilities already present at the FL–2C site. Data were made available to application programs via the group standard *server–client* communications package. Time–averaged LLWAS data from the 15 LLWAS stations in and around the Orlando airport were collected and put in a T–LAPS compatible file format by the LLWAS data input program. The wind analysis program cannot make effective use of observations that map to the same grid point and all the LLWAS stations map to the same 10 km grid point. Therefore, for the 10 km wind analysis, data from all the LLWAS stations were averaged to a single value and written out as a single synthetic station. The individual LLWAS station data were used by the 2 km wind analysis.

The LLWAS data input program updated the data files every 10 seconds, keeping up with the inherent update rate of LLWAS. Every five minutes, when the T–LAPS wind analysis was run, the latest LLWAS data file was combined with the latest surface station data into a single ground station data file that could be read by the wind analysis. The LLWAS data ingest was very reliable. The LLWAS had to be turned on before T–LAPS started; otherwise the LLWAS data input program would time out and abort. A future enhancement is to modify the LLWAS data input program to keep on trying instead of timing out; this will ensure that T–LAPS will get LLWAS data whenever it is available.

3.5. Radar Data

Radar data were ingested using facilities already present at the FL–2C site. Radar data were broadcast into a special Ethernet – the “base–data” Ethernet – dedicated to this purpose. Any machine that needed to read raw radar data had to be connected to this special Ethernet. A second Ethernet was present at site for more routine communication between machines. A third Ethernet was installed to handle NEXRAD data. Lincoln has an extensive library of software to assist in the task of reading this data. One package – *server–client* – pulls the raw data off the Ethernet. A second software package assembles the raw data into radials and further assembles the radials into tilts. Application programs then deal with these tilts of data and do not have to worry about the low–level details of accessing the data.

Weather radar data comes from the radar in a polar coordinate system. Every radar data value, sometimes called a range gate, is indexed by azimuth angle, elevation angle, and range from the radar. A single sweep of the radar at a constant elevation angle is called a tilt. Weather radars usually iterate over a fixed pattern of tilts at various elevation angles. A collection of all the tilts in a single scanning pattern is known as a volume scan. To be usable by the wind analysis this data must be re–mapped from its original polar coordinate system into the gridded Cartesian coordinate system used by T–LAPS. This process is known as resampling.

The T–LAPS update rate of five minutes was chosen because that is the approximate time it takes for a TDWR to complete a full volume scan. However, the TDWR scanning pattern can change and the exact time for a volume scan varies somewhat. Moreover, the NEXRAD has its own scanning patterns with timings that differ from the TDWR's. Since T–LAPS runs on its own rigid five minute cycle, there are synchronization problems.

One approach would have been to write a resampled radar file out every time a radar finished a complete volume scan and have the Analysis module pick the most recent radar file every time it ran. This is the strategy used in LAPS as run by FSL. This approach has the advantage of simplicity, but it has the disadvantage of the radar data used by T–LAPS possibly being several minutes old. Since exploiting the radar's rapid update rate is part of the rationale for T–LAPS, it was decided that this was unacceptable. A better solution was made possible by the fact that radar data is not made up of indivisible volume scans; rather, it is made up of a continuous sequence of tilts. Moreover, the division between volume scans is fairly arbitrary and there was no inherent reason that a selection of tilts that crossed a volume scan boundary could not be used. Therefore, the approach used by T–LAPS was to construct a synthetic volume scan from the most recent tilts. This ensures that T–LAPS is using the most up–to–date radar data possible.

The resampler was split into two parts. The first part was the tilt resampler. This was a totally data–driven piece of software that resampled the tilts as they were read off the base–data Ethernet. This was similar to how other group applications worked, so this was the part of the resampler that was built out of pre–existing group software. The polar tilts were resampled into 2–D grids that match the grids used by the 2 km and 10 km analysis programs. The resulting data was then written out, each tilt in its own file, to a resampler working directory. The second part of the resampler was the volume resampler. The volume resampler was activated every five minutes under the control of the driver. When the volume resampler was activated, it looked in the resampler working directory for the most recent set of tilt files. The contents of the tilt files were then combined vertically and the final 3–D radar file was written out in T–LAPS sparse gridded format.

The tilt resampler was based on a median filter. For T–LAPS it was thought quite important that the radar data be relatively free of contamination from noise and other artifacts, and median filters are good at removing outliers. A 1 km by 1 km median filter was applied to the radar tilt to produce a smoothed tilt. It is simpler to use a median filter that has a fixed width in radials, but the narrowing of radials near the radar means that the spatial width of the filter would be very small near the radar. Instead, a more complex filter that used a variable number of radials in order to maintain a constant spatial filter width was used. With median filters, computation speed can be an important consideration. The sorting operation used in a median filter can be extremely expensive if done naively. To address this problem, a sliding window approach was used which takes advantage of the fact that the set of points to be sorted does not change very much between adjacent radar range gates. The median filter also allowed for the filling in of some missing values if there were a sufficient number of neighboring points around the missing value. The mapping to a 2–D grid was done by a simple lookup algorithm. For each grid point in the 2–D grid, the value of the closest point in the smoothed polar tilt was used as the value of the grid point. This lookup process was repeated twice: once for the 2 km grid and once for the 10 km grid.

The main work of the resampler was done by the tilt resampler. However, the T–LAPS Analysis module expected its radar data to be in the form of a single 3–D grid, with the vertical axis in pressure coordinates. The function of the volume resampler was to combine the individual tilt files into a single 3–D grid. When the volume resampler was activated by the driver, it picked out the set of most recent tilts. Among the tilts with the same elevation angle, the most recent one was kept and the rest were discarded. All tilts older than 10 minutes were discarded. For each grid point in a tilt, the height could be calculated easily from the elevation angle and the range to the radar. After the height in meters was computed, it was converted to a height in pressure coordinates – millibars – using standard height/pressure conversion equations. This introduced a slight inaccuracy into the calculations because the actual pressure levels as represented by the MAPS files would not necessarily exactly match the standard height/pressure conversion. However, the difference was small and not considered important. In this manner, all the points in the working set of tilts were projected into 3–D space. Limited vertical extrapolation and interpolation were used to obtain values at the T–LAPS vertical coordinate levels.

3.5.1. TDWR

Facilities for ingesting TDWR data had already been in place at the Kissimmee FL–2C site since the site was established in 1990. Nothing needed to be added for the T–LAPS project other than the T–LAPS computers. The TDWR data came to T–LAPS with velocity unfolding, range obscuration editing, and point–target editing already done using standard TDWR algorithms. The data was not edited for signal to noise ratio (S/N), but a S/N product was available, so low–signal data points were edited out using a S/N threshold of 0 db.

FL–2C is sited near the Orlando airport – the center of the T–LAPS domain – and the size of the T–LAPS domain was chosen with the TDWR scanning range in mind, so the T–LAPS domain is a good match with the scanning area of FL–2C. Thus, the resampler needed to process all of each TDWR tilt, putting quite a strain on the available computer resources. For both algorithmic reasons and to provide extra time to resample NEXRAD data, TDWR tilts over 45 degrees elevation were not resampled. Due to computer capacity restrictions, reflectivity was resampled only on the low–angle tilts. Reflectivity information was not used by the analysis system, so the low–angle reflectivity tilts were resampled solely for display purposes. Figure 10 shows the relationship between FL–2C, the Orlando airport, and the rest of the T–LAPS domain. The Orlando airport runways are represented by the short vertical lines in the center of the figure. The location of FL–2C is shown by the dot just south of the airport.

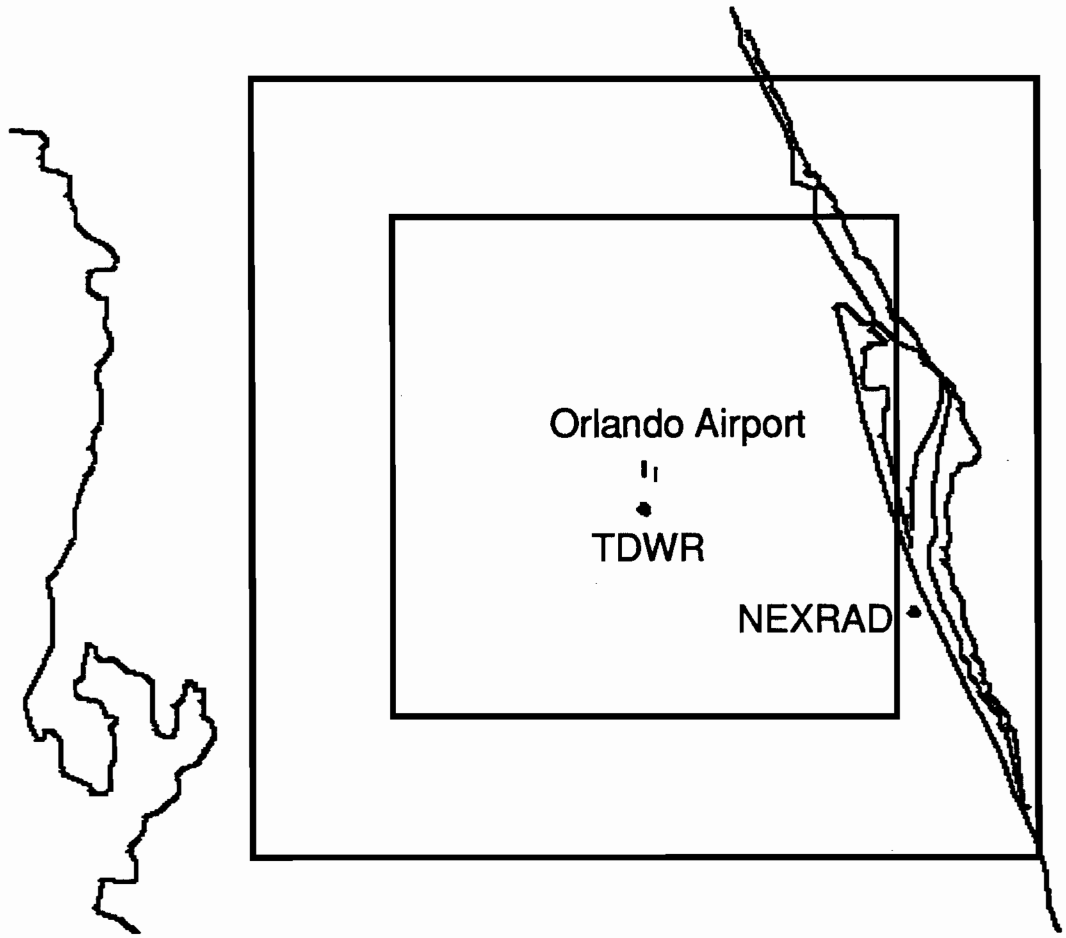

Figure 10. T–LAPS domain showing radars.

3.5.2. NEXRAD

NEXRAD/WSR–88D data was brought into the FL–2C site using a T1 leased line from the NEXRAD site in Melbourne, Florida. This was a new installation for the summer of 1992. To distribute the NEXRAD data to the machines at the site, a second base–data Ethernet – the third Ethernet overall – was installed at site. Application software accessed NEXRAD data in exactly the same manner as TDWR data, except that the relevant machine had to be connected to the new Ethernet.

As with the TDWR, tilts of elevation higher than 45 degrees were discarded and the reflectivity product was resampled only for low–angle tilts. The NEXRAD data were edited for range obscuration, but not for velocity unfolding. The large NEXRAD Nyquist interval (26 m/s) and the scarcity of severe weather during the summertime in Florida (hurricane Andrew notwithstanding) meant that velocity folding was rarely a problem. NEXRAD data was edited for S/N by the NEXRAD computer. The exact value of the threshold is a variable site parameter. The NEXRAD data does not come with a S/N product, so it was impossible to re–edit using a higher S/N threshold.

The NEXRAD is sited near the eastern coast of Florida, to the south of Orlando. It is is a few kilometers off the eastern edge of the 2 km domain but is within the 10 km domain. Its location is shown in Figure 10 by the dot just outside the eastern edge of the 2 km domain. The NEXRAD has a scanning range of 230 km for velocity and 460 km for reflectivity. Therefore, the total NEXRAD scanning area is much larger than either T–LAPS domain. For reasons of speed, it was important to avoid median filtering an entire NEXRAD tilt. The NEXRAD tilt resampler is designed to median filter only the portion of the tilt that falls within the T–LAPS domain. Because of this screening, and because the NEXRAD range gate resolution is coarser than TDWR, the NEXRAD resampler was much faster than the TDWR resampler. This was important because the TDWR resampler alone took up much of the capacity of the resampling machine.

4. MAIN ANALYSIS PROGRAMS

The two sub–components of the Analysis Module, the wind analysis package and the balance package, were both built out of code that was originally written at FSL. This section will describe the porting process of moving the code from the FSL system to the Lincoln Laboratory computing environment and document the changes that were made to the original FSL code. The changes can be classified into two categories. The first category is the porting changes needed to get the FSL software to work in the Lincoln computer environment. The second category is the algorithm changes needed to make the system work better in the T–LAPS context.

4.1. Porting the FSL Software.

The FSL computing environment that was used to develop the LAPS code was based on VAX/VMS computer systems. The source code was written in Fortran 77, with heavy use of VMS extensions. Rewriting the code to conform with the Fortran 77 standard would have been a very large task. In addition to the Fortran extensions, the code made assumptions about the surrounding VMS environment. File names of both data files and "include" files used VMS syntax. VMS library routines and VMS system calls were used for some aspects of file handling, and VMS system calls were used to keep track of time stamps. The difficulty of the job was somewhat alleviated by the fact that the code was a modest 30,000 lines long (including comments) and the really essential code was only about 15,000 lines long.

The target computing environment was a network of Sun workstations running SunOS 4.1.1. SunOS is a descendent of Berkeley Unix. The Suns were equipped with a Fortran 77 compiler which – fortunately – accepted many of the VMS Fortran extensions. However, none of the VMS system calls were supported, and Unix pathname syntax is quite different from VMS pathname syntax. SunOS comes with the usual Unix suite of utilities for manipulating text files; some of these tools were used to semi–automate the porting process.

A complicating factor of the porting task was that the code was a moving target. While the porting task was underway, FSL was adding upgrades to the code, especially to the wind package. Scheduling pressure and the fact that Lincoln Laboratory was actively involved in designing and testing some of the upgrades precluded the option of waiting until FSL was finished with their changes and porting the code once. Therefore, the porting procedure had to be carefully designed to be repeatable so that new changes to the code could be brought across with little effort.

The most important part of the strategy was to make portability changes to the LAPS software and then send these changes back to FSL for them to incorporate in their code base. As this process continued, the code set being maintained by Lincoln Laboratory and FSL grew more similar to one another, which eased the task of integrating new code. A key element of this strategy was the active cooperation of FSL. FSL was always very quick to respond and was usually quite receptive to Lincoln's request for code changes. FSL had their own long–range plans to move the LAPS code to Unix, so they also benefited from this cooperative effort.

A semi–automated procedure was worked out for handling upgrades to the LAPS code. The first step was to save unmodified copies of the original FSL source code in a special directory. After editing the files in the regular working directory, the Unix `diff` utility was used to automatically record all differences between the new Lincoln Laboratory version and the original FSL version and save them to a "diff" file. This had a useful bookkeeping function because the set of "diff" files made up a complete record of all departures from the FSL code. When a new version of a file was received from FSL, the Unix utility **patch** was used to quickly re–insert the Lincoln Laboratory changes as recorded in the "diff" file back into the FSL file to produce a new Lincoln version. As long as the Lincoln changes and the FSL changes were independent, all was well. However, if Lincoln and FSL had modified the same lines of code in a file, a conflict would arise that `patch` would not be able to resolve automatically. In these cases, the conflict was resolved either manually or with the help of the Unix Revision Control System (RCS) program, **merge**.

4.2. Specific Porting Changes

4.2.1. Pathnames

The most pervasive problem was the difference between VMS pathnames and UNIX pathnames. The pathnames differed due to differences in pathname syntax between the two operating systems as well as the different organization of the FSL and Lincoln Laboratory systems. The problem could be divided into two categories. The first was the problem of the "include" pathnames embedded in the source code. The second involved the pathnames that the system used when reading or writing data files.

Solving the problem of the "include" pathnames was the easier of the two problems. Because of the inherent nature of "include" files, the names of files were static and there was no manipulation of the files at run time. A simple **sed** script was used to remove all VMS directory information from the VMS pathnames that were in the code. The pathnames that were left were now valid UNIX pathnames, albeit without any directory information. Since the ported FSL code was placed in only in a small number of source directories it was a simple matter to copy the proper "include" files into the directories where they were needed.

A much more difficult problem was dealing with the pathnames that the system used when reading or writing data files. The procedure described above was a partial solution, but it left us with a system able to read and write data files only out of the current working directory. Moreover, there was still some FSL code in the system that manipulated pathnames based on VMS rules. It was considered desirable to be able to specify the data directories at run time from the command line, so additional functionality had to be added. FSL already had a routine that returned a directory based on a file name's type extension – get_directory – so that, for instance, the radar data files could be put in a different directory from the surface station files or the ACARS files. This was very useful to FSL because of the way their system was organized, but the manner in which the directory names were embedded directly in the code made the routine too inflexible for the Lincoln Laboratory computing environment. It was fairly simple to re–write get_directory so that it used directories specified on the command line. However, get_directory had not been used uniformly in the FSL code – some sections of the system relied on hardcoded directory names – and it was quite a bit of work to made sure that get_directory was being used in all possible places. Additionally, some routines that manipulated pathnames had to be carefully re–written so that they worked with either Unix or VMS pathname syntax. For example, a routine to separate out the directory component of a pathname had to look for both the "]" character for VMS and the "/" character for Unix. Of course, completely separate routines could have been written, but it was desirable to keep the Unix version and VMS version as similar as possible.

4.2.2. Logical vs. Integer Variables

The most difficult problems were caused by confusion between logical variables and integer variables. Many Fortran dialects are fairly permissive about assigning logical values to integer variables, but the practice is non–standard and not portable between machines. The exact value of some expressions depends on the details of the machine representation of logical values. Testing exposed several coding practices that apparently worked on a VAX/VMS system but did not work on Suns. For example, the following code compiles and runs without error messages or warning messages, but it prints "A" *and* "B" when run on a Sun. This is not an error on Sun's part because the use of an integer variable in a logical expression is not standard Fortran.

```
integer i
i = 1
if (.not. i) print *, 'A'
if (i) print *, 'B'
```

These problems were difficult to find initially, because of the lack of any obvious indicator that anything was amiss. The FSL code had a consistent pattern of applying logical tests to integer status variables. After the problem was identified, it was a simple, although rather tedious, task to search all the FSL code for expressions of this nature.

4.2.3. VMS Time Representation

Both Unix and VMS have a method of representing time as a single integer that is the number of seconds since a fixed reference date. Unfortunately, Unix uses January 1, 1970 as its reference date, while VMS uses January 1, 1960 as its reference date. To complicate matters, the FSL code used VMS library routines to help manipulate times and to generate ASCII representations of times. Unix library routines could not be used as replacements partially because of the reference time problem and partially because the Unix library routines were not portable to the VMS system. To deal with this problem, a set of portable Fortran routines were developed that performed all the needed time/date manipulations without any system calls. The work was facilitated by adapting routines that had already been written for the FSL Unix version of the balance package (see section 4.4).

4.3. Wind Analysis Package

While the porting project was underway, FSL modified the wind package to implement the multi–single Doppler algorithm described briefly in section 2.2.1. That was the biggest single change, but there were quite a few smaller changes that had to be made in order to adapt the LAPS wind analysis code to the requirements of the T–LAPS system. The real difference between the 2 km wind analysis and its LAPS predecessor is the central role of radar data in the T–LAPS version. The really obvious changes to the software – the smaller domain size, the finer resolution, and the faster update rate – are all reflections of characteristics of the main TDWR data source. Moreover, there were a number of smaller changes made to the software to adapt it to the demands of the T–LAPS environment. Some changes had to be made simply to adapt the system to the gigantic number of observations that a weather radar can produce.

4.3.1. Processing of Radar Data

There were several relatively minor changes regarding the handling of radar data. Data from the TDWR comes to T–LAPS already velocity unfolded, but this is not true of the data from the

Mile–High radar – FSL's source of radar data. NEXRAD did not come to T–LAPS unfolded, but as mentioned previously, velocity folding was rarely a problem. Thus, T–LAPS was able to be less strict about radar data quality control than was the original FSL system.

The original FSL code applied a quality control check on the radar observations by comparing them against the background field. Where there was a radar observation, the radial component of the background field with respect to the radar was computed. If the difference was greater than a threshold, the the radar observation was discarded. The original FSL threshold was 12 m/s. In T–LAPS '92 the threshold was increased to 20 m/s.

The original FSL code attempted to velocity unfold the Doppler data by comparing it against vertical profiler data. If there was no profiler data, then all the radar data would be thrown out. As T–LAPS was not using any profiler data, this obviously had to be changed so that radar data could be used. The change was not difficult because there was a pre–existing parameter in the code that disabled the profiler dependency. FSL changed the '93 version of LAPS so that the output of the first Barnes pass is used for velocity unfolding instead of profiler data. This is much more general because the first Barnes pass combines all vector observations, including profiler data if available, instead of just profiler data alone.

Due to the computational demands of interpolating large numbers of radar observations, the LAPS wind analysis adopts a strategy of deliberately throwing some of the radar data away. When there is a large amount of radar data on a level, the wind analysis throws out three out of every four radar observations. Special care is taken to retain isolated observations. If there were only a few radar observations on the level, then the radar observations were not thinned out. The original FSL threshold was 150 radar observations. In T–LAPS the threshold was raised to 400. An important consequence of the increased limit was that radar data was never discarded in the 19 by 19 10 km T–LAPS wind analysis because there were only 361 grid points per level.

4.3.2. Barnes Interpolation

In the original 61 by 61 10 km LAPS wind analysis, the output of a TDWR would fill only a relatively small portion of the overall grid. This still could be several hundred observations per layer, which was much larger than the number of observations from other data sources. In the 19 by 19 10 km T–LAPS wind analysis, the picture was similar; there were never more than a few hundred radar observations per layer. At the 2 km grid scale, the picture changes dramatically. By design, the radar data had the potential (depending on the amount of significant weather) to fill almost the entire grid with observations. The maximum number of observations per layer rose to several thousand. Even after the radar data were thinned, there could be nearly a thousand observations per layer. The maximum number of radar observations from a single radar in one volume scan was observed to be in excess of twenty thousand. Due to this load, the Barnes interpolation had to be reworked to make it more computationally efficient. All the changes to the Barnes interpolation described below were adopted by FSL.

The original FSL implementation of Barnes interpolation placed all the observations for all layers into a single temporary array that could hold up to five thousand observations. If the five thousand limit was exceeded, the interpolation routine would abort and the final result would be a field of zeros. The Barnes routine was changed to work on a purely layer–by–layer basis. The temporary array in question now only had to hold the observations for a single layer at a time. The new array size was set at four thousand so that it would be guaranteed to be larger than 61 x 61, the maximum possible number of observations per layer (multiple observations per grid point are not allowed). An additional benefit of this change to the Barnes routine is that the code can now be run more efficiently on a parallel computer.

The execution speed of the wind analysis was heavily dependent on the speed of the Barnes interpolation. The speed of the Barnes interpolation was roughly proportional to the total number of observations. During average conditions with little weather and few radar observations, the Barnes interpolation was very fast. When there was lots of significant weather in the Orlando area and thus many radar observations, the speed of the Barnes interpolation slowed dramatically to the point where the time to calculate the 2 km wind analysis exceeded the five–minute cycle time. Two changes were made to the Barnes routine to speed it up. Initially, the **u** and **v** components of the wind field were being computed separately. This helped make the Barnes routine more general – it could be used to interpolate a scalar field such as pressure or temperature – but it wasted time when calculating winds because the calculation of internal interpolation weights was always identical for **u** and **v**. Whenever there was a **u** observation, there was always a **v** observation, and vice versa. Thus, for any target grid point, the weights associated with interpolating a **u** value were always identical to the weights associated with interpolating a **v** value. The code was changed so that the **u** and **v** passes were combined into a single pass with a single interpolation weights calculation. This change made Barnes take about 70 percent of the time it did previously for cases with large numbers of observations. A second change was to eliminate a call to the Fortran intrinsic function `nint` in the inner loop of the code. The compiler was not able to fully translate the call into direct machine instructions, so there was a function call to support the subroutine in the inner loop. Additionally, the `nint` routine had more generality than was needed – it was not able to take advantage of the fact that, given the way the call was used in the interpolation routine, its argument was always positive. The `nint` call was replaced with an expression that used Fortran implicit conversion rules. This was compiled into a single machine truncate instruction. All told, the speed of the Barnes interpolation routine was more than doubled and the overall speed of the wind analysis was approximately doubled for high–load cases.

There was another change to the Barnes interpolation routine that was unrelated to the speed enhancements described above. As described later in Section 5, the driver took over the role of interpolating MAPS data. A quirk of the Barnes code as originally written was that it used some data structures that were tied to the exact details of how MAPS was handled by the wind analysis. This meant that the Barnes interpolation was inadvertently changed whenever the treatment of MAPS data changed. This was obviously undesirable, so the Barnes interpolation code was modified so that its data structures were independent of any MAPS–related data structures.

4.3.3. Miscellaneous Changes

The original FSL winds package discarded ACARS observations that were older than 60 minutes. The size of the T–LAPS domain (180 km by 180 km) was considerably smaller than the size of the Denver LAPS domain (600 km by 600 km), so T–LAPS received fewer ACARS reports than did LAPS. Since ACARS observations were the only observations that provided data above where the radars could see, the time window was extended to 90 minutes for T–LAPS. Additionally, the time windowing functionality was moved out of the wind analysis code and moved into the driver. The driver also handled the time windowing for some of the other data sources, such as surface stations, so the change centralized the time windowing functionality.

The original FSL winds package calculated the "time tendencies" of the MAPS data by comparing the two MAPS datasets nearest in time to the current time and computing the rate of change over time. The resulting MAPS time tendencies were then used to adjust the ACARS reports, because the ACARS reports could be relatively old. The structure of the T–LAPS system, with the driver doing all the manipulation of the MAPS files, made it somewhat inconvenient to implement this functionality in T–LAPS. The adjusting of ACARS reports was thought to be a somewhat experimental technique that was not necessary for T–LAPS '92. The code that attempted to read in MAPS files to compute the time tendency field was disabled and the time tendency field was set to zero. This had the effect of no longer adjusting the ACARS observations without requiring large code changes.

After the horizontal **u,v** winds were computed, the original FSL winds package then attempted to compute a vertical motion field. This was done by estimating the convergence/divergence in the horizontal winds and adding in a terrain forcing component. T–LAPS did not make use of the vertical motion product and writing it out took a large amount of space in the winds output file. The **u**, the **v**, and the **w** (vertical motion) products all took up an equal amount on space, so omitting **w** made the wind output files 33 percent smaller. This change also saved a small amount of compute time, although that was not a major consideration.

After the primary wind analysis was performed, several secondary wind products were calculated by the original FSL winds package. Radar reflectivity information was read in and used to create several products. A map of the heights of reflectivity maximums was created. The winds between the surface and 300 mb were averaged to create a mean winds product. A simple storm tracking algorithm was then used to adjust the mean winds. A map of maximum reflectivity was created and then the reflectivities were advected using mean winds to produce a forecast. None of these outputs were important for T–LAPS '92, so all of these calculations were deleted from the T–LAPS version of the wind package.

4.4. Balance Package

When the T–LAPS project began, FSL had already ported an older version of their balance package to a Stardent running Unix. The Stardent version of the balance package proved to be relatively easy to port to our target Suns. It was a self–contained package that came complete with its own set of utility routines. The most up–to–date version of the balance package, however, was written for VMS and had not been ported to the Stardent or any other Unix machines. Fortunately, the code that implemented the balance package core algorithm was only a few thousand lines of code long and it contained relatively few VMS dependencies. The port of the new version to the target Suns was not difficult because the new code that implemented the balance algorithm was successfully mated with the utility package that had been provided with the Stardent version. It also helped that the FSL balance package underwent no changes in the time period preceding the 1992 deployment.

No changes were made to the balance package to add any important functionality. There were a number of small changes made to adapt the system to the T–LAPS environment. Most importantly, the code assumed a 61 by 61 grid, while the 10 km T–LAPS analysis used a 19 by 19 grid. It was also desirable to be able to continue to use the code in 61 by 61 mode in order to compare outputs with FSL. This was not terribly difficult, but the code had not originally been written to support multiple grid dimensions simultaneously. A few minor changes were made to adapt the system to running on a 30–minute cycle instead of a one–hour cycle. The code also had to be altered to adapt to the absence of expected data sources that were not generated by the T–LAPS system. The balance package expected two different estimates of the vertical component of wind motion. One estimate was derived from the national MAPS grid; the other was calculated by a cloud analysis module that is a part of FSL's LAPS. By default, the balance package would abort if neither of these data fields were present, so balance was modified so that it did not abort, and zeroed out the vertical motion field instead. The balance package relied on a surface pressure field to locate its lower boundary conditions in pressure coordinates. The surface pressure field is calculated by the LAPS surface analysis package, which was not used, so code was added to the balance package in order to fill the surface pressure field with a constant 1000 millibars. Since each LAPS vertical layer was 50 millibars thick and surface pressures would not be be expected to vary much over the Orlando domain, this was a reasonable assumption.

While the balance package was being tested, several bugs were found in the code, with at least one of them serious. The Sun Fortran compiler supports array bounds checking (this feature is quite common among Fortran compilers); the use of this feature allowed the detection of several array bounds overflow bugs in the FSL code. In several places in the code, invalid data was being accessed and used in the algorithm. This was less catastrophic than it sounds; the balance package usually produced plausible looking output. When FSL fixed the balance package bugs in their own code, a before/after test showed differences of up to 5 meters/sec in the wind values at some grid points.

5. REAL–TIME DRIVER

The real–time driver module oversaw the operation of the T–LAPS '92 system. The driver was the timekeeper of the system. It scheduled and coordinated the operations of other modules. The driver was a new software module developed at Lincoln Laboratory. It was developed because the driver used by FSL for their LAPS system was written for the VMS operating system and was designed to use FSL–specific data ingest methods. Furthermore, the T–LAPS grid sizes and the 10 km/2 km cascade approach were different from the FSL LAPS system.

The driver performed the system start–up and scheduled data acquisitions, winds analyses and product displays. The driver also prepared the data for analyses, performed data archiving, and coordinated system shutdown. The design of the driver included the built–in flexibility to handle erroneous conditions such as missing data, late data, failed processes, empty data files, illegal file formats, etc. In preparation of the backgrounds for the T–LAPS analyses, data interpolations in both spatial and temporal domains were performed. Since the driver was responsible for coordinating processes running on multiple machines, a simple signal file technique was employed for interprocess communication.

To start operations, the driver first started up a number of independent asynchronous programs such as the resamplers and then used the MAPS data to initialize the 10 km wind and pressure/height backgrounds. After initialization, the driver went into an infinite loop with a cycle time of five minutes. During the loop, the driver had a list of events to run. It kept track of the current time and invoked various data collection programs at certain fixed time intervals. The T–LAPS 10 km analysis was invoked every 30 minutes and the T–LAPS 2 km analysis was invoked every five minutes. A valid data time–window was used to ensure that only recent data was used in the analysis. It was the driver's responsibility to make sure that all the data was inside the valid data time–window. The driver was also responsible for archiving the data files for later analysis and playback. During the five–minute loop the driver moved the data files to the archive directory as soon as the data files were no longer needed.

The timing requirements of the driver were not very stringent. There were no major consequences if the driver fell behind in one of the 5 minute loops. The next loop would start a little late, but it would eventually catch up. During the T–LAPS '92 demonstration, no major bottlenecks other than the wind analysis were observed. The driver skipped the 2 km wind analysis if the processing fell too far behind. However, the data collection portion of the program was always executed such that later analysis and playbacks were not affected. The driver occasionally skipped the 2 km wind analysis at the early stage of the T–LAPS '92 demonstration. After various optimizations were implemented in the wind programs, the system rarely skipped the 2 km wind analysis, even under heavy weather conditions.

The T–LAPS '92 system was implemented on two dedicated Sun SPARCstation 2 workstations at the FL–2C site. The two workstations were connected via the network file system (NFS). One of the SPARCstations was dedicated to the radar resamplers. The other workstation hosted the driver, the dial–up programs, the wind analysis packages, the balance package, and the display. Figure 11 illustrates the physical distribution of the processes of T–LAPS '92.

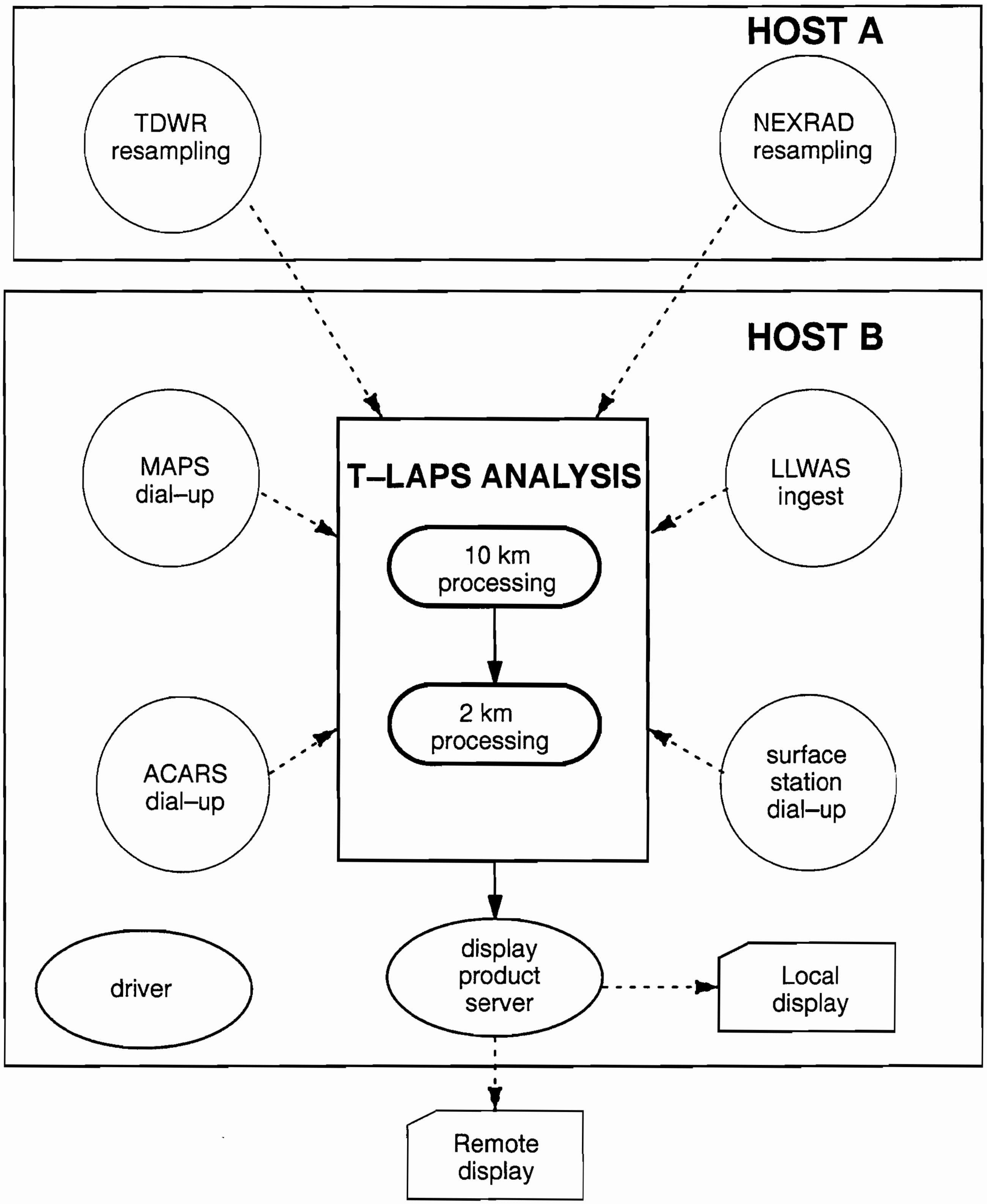

Figure 11. T–LAPS processes and hosts.

5.1. System Start Up

Shell scripts were used to start all the T–LAPS independent processes which included:

1. The TDWR resampler program
2. The NEXRAD resampler program
3. The MAPS dial–up program
4. The surface station data dial–up program
5. The ACARS dial–up program
6. The LLWAS ingest program
7. The display product server program
8. The driver program

5.2. Interpolation of Wind Analysis Background

The MAPS data from FSL was used to provide the background information for the T–LAPS processing. As explained in section 3.1, MAPS three–hour and six–hour forecasts were interpolated to construct the synthetic MAPS datasets for the 10 km processing. The interpolation was required in both the spatial and temporal domains.

5.2.1. Spatial Interpolation

The MAPS data received from FSL was a set of 6 by 6 horizontal grid points with 21 vertical layers for the central Florida region. MAPS used a polar–stereographic coordinate system that was aligned on Denver's longitude. As a result, the MAPS grid was tilted whenever the area of interest was not at the same longitude as Denver. The spatial interpolation was performed by a bilinear interpolation algorithm, but the fact that the MAPS grid and T–LAPS grid were not aligned with each other complicated some of the details. Figure 12 shows the alignment of the MAPS grid points compared with the T–LAPS domain.

In order to make the interpolation more efficient, tables of interpolation coefficients and MAPS grid indexes were computed beforehand. For each 10 km grid point, the tables held the indexes of the four surrounding MAPS points and the interpolation coefficients for the four MAPS points. During real–time operations, the interpolation was done at initialization time and thereafter for each new set of the MAPS data. The process of interpolation first read in the four index points and the interpolation coefficients from the tables for each of the 10 km grid points and then added together the multiplied product of the weight coefficient and the value associated with the MAPS index point.

5.2.2. Temporal Interpolation

For every set of new MAPS data, the MAPS three–hour forecast was interpolated spatially to the 10 km grid. The difference between the three–hour forecast and the six–hour forecast was also computed and spatially interpolated to the 10 km grid. Using these pairs of spatial interpolation results, the 10 km background for any arbitrary time could be obtained by linear temporal interpolation.

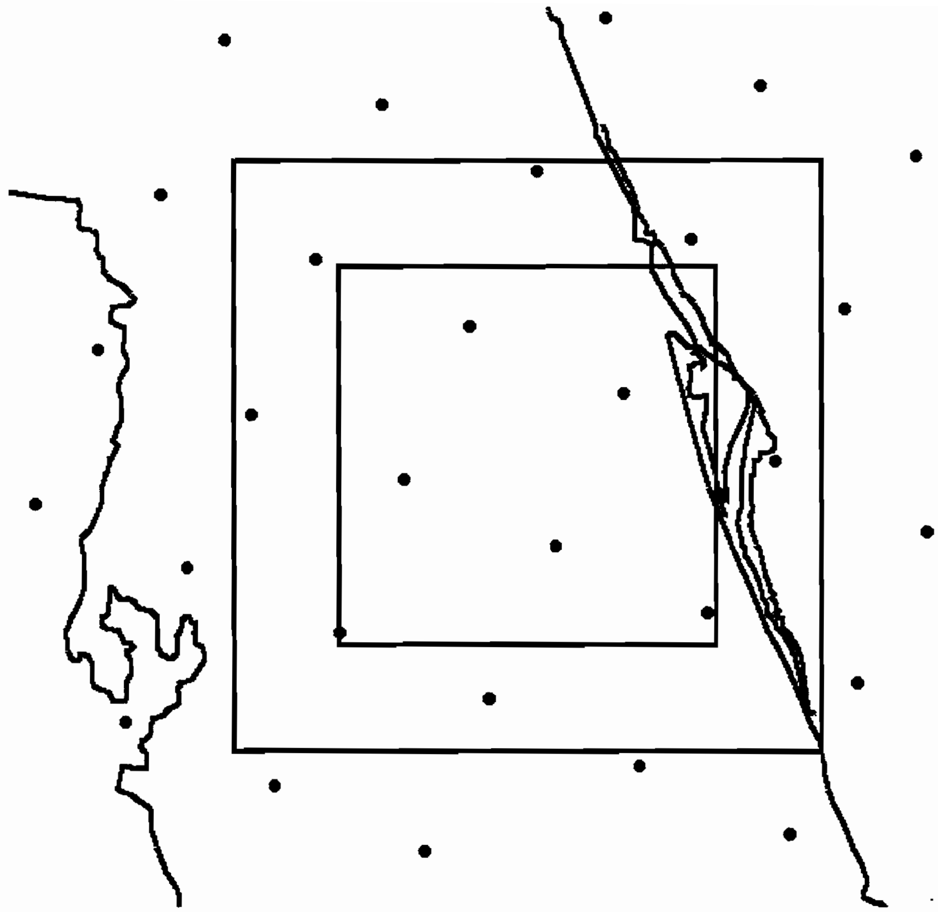

Figure 12. MAPS grid points.

5.2.3. Height Adjustment

The MAPS subset used by T–LAPS '92 contained the three products needed by T–LAPS: **u** winds, **v** winds and heights. Each MAPS layer was 50 millibars apart. The heights field gave the altitude of each pressure layer. The spatial and temporal interpolations were performed on all three of the MAPS products. The T–LAPS system used the pressure reading of the MCO surface station located in Orlando, Florida to fine tune the interpolated height field. The closest 10 km grid point to the MCO station was computed and the MCO height was calculated from the interpolated height product using the MCO pressure reading. The offset between the calculated MCO height and the real MCO height was used to adjust the height product.

5.3. T–LAPS 10 km Processing

"10 km processing" refers to the 10 km analysis plus various utility programs such as data ingest programs. The 10 km processing was performed every 30 minutes. Most of the surface stations were updated hourly on the hour, so T–LAPS dialed up the NASA computer to obtain the data at five minutes after the hour and every 15 minutes thereafter. The 10 km processing was scheduled at 10 minutes and 40 minutes after the hour in order to include the latest possible surface station data. A detailed flowchart of the T–LAPS 10 km processing is shown in Figure 13.

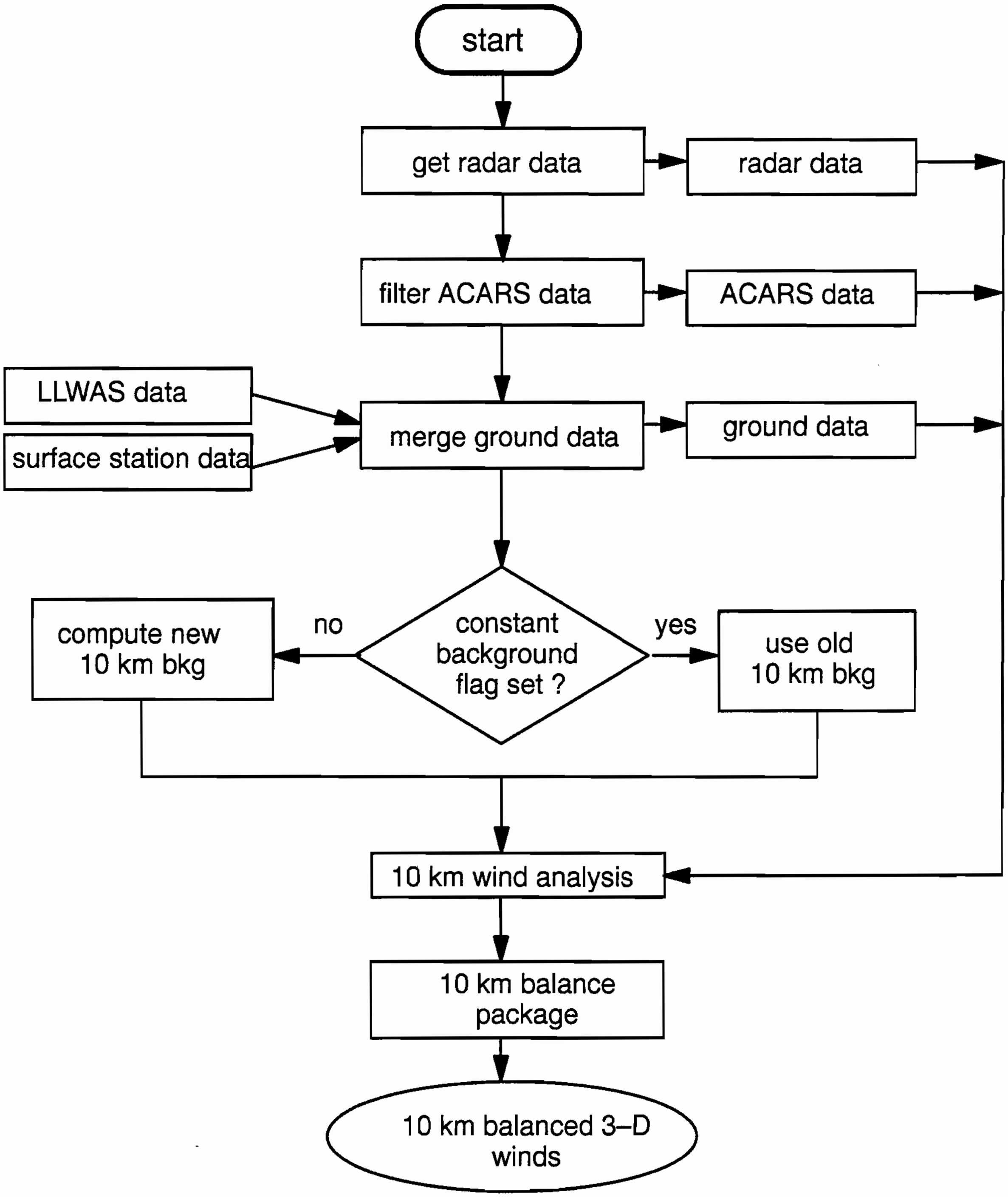

Figure 13. T–LAPS 10 km processing.

5.3.1. 10 km Data Ingest

There were six types of input data used by the 10 km processing–MAPS data, TDWR radar data, NEXRAD radar data, ACARS data, surface station data, and LLWAS data.

The initial dial–up for MAPS three–hour forecast data files was done at two hours and 30 minutes into the MAPS cycle and the initial dial–up for MAPS six–hour forecast was done at three hours into the MAPS cycle. If the dial–up failed, the process was repeated until it was successful, as explained in section 3.1. The MAPS three–hour forecast and six–hour forecast data were interpolated spatially and temporally, and they were used as the background for the 10 km analysis. In the case where MAPS data was not available, a constant background of zero winds was used (see section 5.7).

The TDWR and NEXRAD radar data were resampled to the 10 km grid. On a signal from the driver, the latest resampled tilt files were assembled into volume data files for each 10 km cycle. The radar resamplers had a built–in data screening function which ensured that only data less than 10 minutes old would be used.

The dial–ups for ACARS data and surface station data were done at five minutes after the hour and every 15 minutes thereafter. There were built–in retries if the dial–up did not succeed. The 10 km and the 2 km processing used the latest available ACARS data files and surface data files at the time of processing. The ACARS data files usually arrived a few minutes before 10 km processing and contained the aircraft reports of the most recent 90 minutes. The 10 km processing read the latest ACARS file and used all the aircraft reports which were less than 90 minutes old. The surface station data usually was obtained from NASA a few minutes before the 10 km processing. The LLWAS data was updated automatically every ten seconds, so the driver simply requested the latest value. All the surface station data of the most recent 90 minutes were combined with the 10 km LLWAS data to form the ground station files for 10 km processing.

5.3.2. T–LAPS 10 km Analysis

The 10 km analysis consisted of two parts. First, the 10 km wind analysis was performed to produce the gridded 3–D wind file, then the 10 km balance package was executed to adjust the pressure and wind fields and produce the balanced 3–D wind product.

5.4. T–LAPS 2 km Processing

The 2 km processing was performed in five–minute intervals on the hour and every five minutes thereafter. At 10 and 40 minutes after the hour, the 10 km processing was performed first, and the 2 km processing was delayed until the 10 km processing ended in order to use the latest 10 km results as the background for the 2 km processing. A flowchart of the T–LAPS 2 km processing is shown in Figure 14.

5.4.1. 2 km Data Ingest

Five types of data were used by the T–LAPS 2 km processing – T–LAPS 10 km results, TDWR radar data, NEXRAD radar data, surface station data, and LLWAS data. The ACARS data was not used because of the long time latency of the data.

The balanced 3–D wind products generated by the T–LAPS 10 km analyses were interpolated spatially to the 2 km grid and used as the background for the 2 km processing. There was no temporal

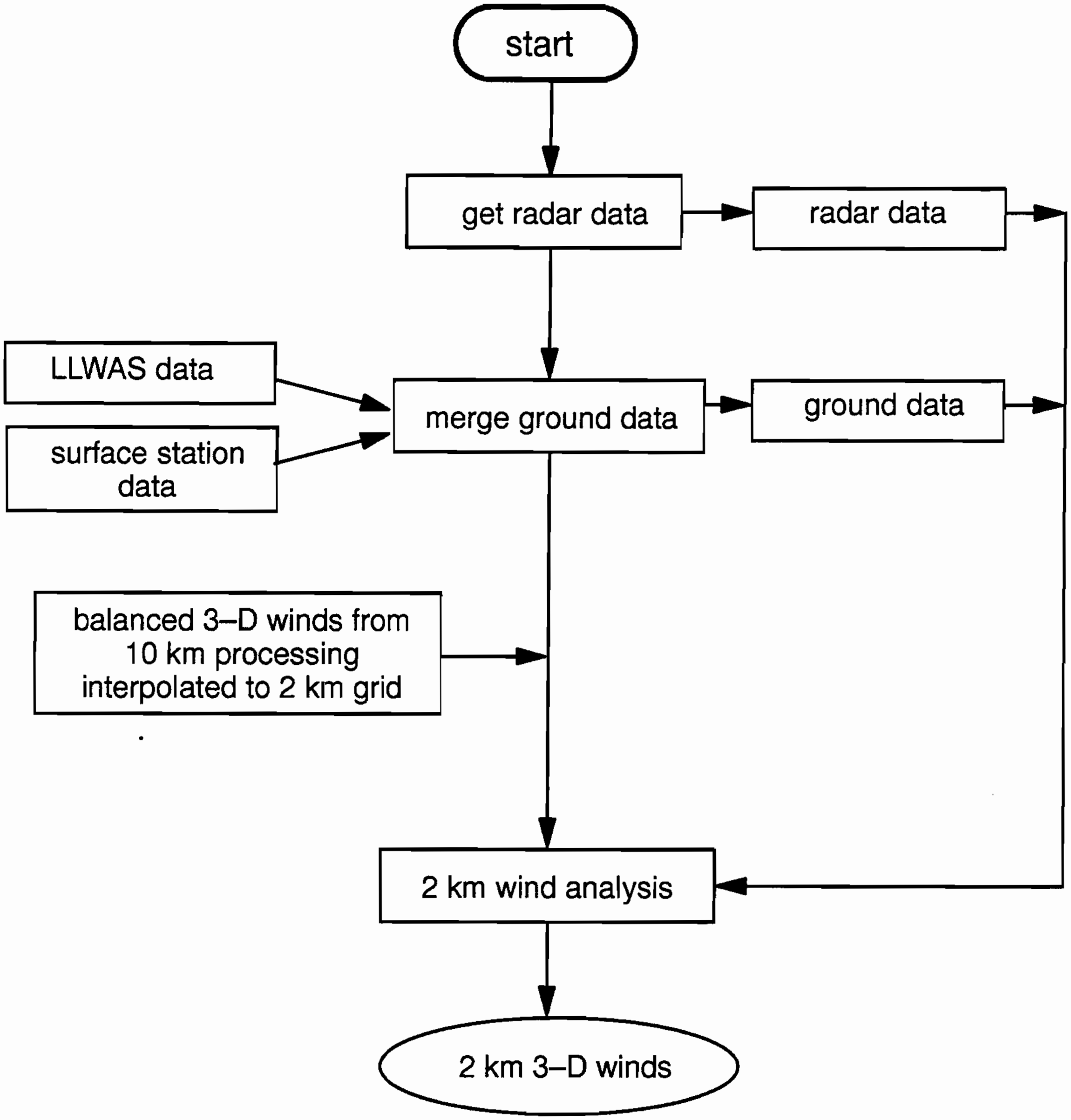

Figure 14. T–LAPS 2 km processing.

interpolation. The 2 km background remained unchanged for 30 minutes until the next 10 km analysis.

The TDWR and the NEXRAD radar tilt data were resampled to the 2 km grid in the same fashion as with the T–LAPS 10 km analysis. The latest resampled tilt files were assembled into the volume data files for each T–LAPS 2 km cycle.

The LLWAS data input program updated the temporally averaged individual–station T–LAPS 2 km LLWAS file every 10 seconds. The surface station data of the past 25 minutes were combined with the latest LLWAS data to form the ground station data file.

5.4.2. T–LAPS 2 km Analysis and Display

The 2 km wind analysis was performed for each T–LAPS 2 km processing cycle. As mentioned in the beginning of this section, the driver would bypass the 2 km analysis if the system was falling too far behind schedule. However, this situation rarely occurred. At the end of the T–LAPS 2 km analysis, the display package was notified to display the newly created T–LAPS 2 km 3–D wind product. The details of the display package are discussed in section 6.

5.5. Data Archive

The driver archived important data files by moving the data files into special archive directories at the end of each processing cycle. The intermediate files which were not needed for later evaluation and playback purposes were deleted. The number of files in the working directories was kept to a minimum for efficiency considerations. The driver also recorded the events of the real–time operations in a log file. The log files were moved to the archive directory when the T–LAPS was shut down each night. The data files were transferred from the archive directories to magnetic tapes at the end of the day for long–term storage. The details of the tape archive process are described in section 7.

5.6. Driver Communications with Independent Programs

The driver program was required to communicate with the radar resampler programs, various dial–up programs, and the display package. Because the T–LAPS '92 system was running on multiple machines (two SPARC workstations), some interprocess communication schemes, such as shared memory semaphores, would not work. The two workstations were connected by the NFS, so the files on both machines were accessible to programs running on either machine. The timing requirements of the T–LAPS system were not very stringent, so a simple signal file interface scheme was used as the interprocess communication method. Due to the file–based structure of the original LAPS system, most modules of the T–LAPS system were already exchanging data files, so adding a few signal files was not much of a burden. The driver program created different signal files to communicate with different programs. For example, if it was time to dial up for the ACARS data, the driver created a specific signal file recognized by the ACARS dial–up program. The ACARS dial–up program was constantly looking for this signal file; once it sensed the existence of the signal file it deleted the signal file and started the dial–up process. Parameters also could be passed in the signal file if needed. This approach was simple, reliable, and had the flexibility to accommodate the processes running on multiple machines.

5.7. Missing Data Conditions

During the operational period of the T–LAPS '92 demonstration, most of the data sources were reliable, but occasionally there were problems in acquiring some of the data. Furthermore, the operational hours of the TDWR radar were shorter than those of T–LAPS. The T–LAPS system was designed with the flexibility to operate under conditions when some, or all, of the data sources were missing. However, the quality of the analysis does degrade under these conditions. When there was a problem in accessing the data sources, the system notified the operator by displaying a message on the console and ringing the bell. This condition also was recorded in a log file. In case there were problems in obtaining ACARS, LLWAS, TDWR, NEXRAD or surface station data, the T–LAPS system continued to operate using the old data as long as the data was within the valid data–time window. Otherwise, the T–LAPS system ignored that particular kind of data.

When the MAPS data was not available to start T–LAPS, synthetic MAPS data with zero **u**, **v** winds and standard pressure–heights were used as the initial background to start T–LAPS. There were a few occasions that the MAPS data were not available in the middle of T–LAPS operations. Under these conditions, T–LAPS continued with the previous background. As soon as MAPS data became available again, the new MAPS background was used. A better approach is to use the LLWAS network mean winds as the initial background and the previous 10 km result as the updated background. This improvement was implemented in T–LAPS '93.

5.8. System Shut Down

A shell script was used to gracefully shut down the T–LAPS system. This script issued kill signals (`SIGKILL`, signal 9) to all the T–LAPS processes which did not need to execute their own clean–up procedures. For those T–LAPS processes which needed to run clean–up procedures, the software termination signal (`SIGTERM`, signal 15) was issued instead. The individual processes could catch the signal and perform last minute cleaning up. The driver archived the log file and the remaining data files of interest in the working directories before it exited.

6. DISPLAY

The T–LAPS display provided the primary interface to the results of the T–LAPS analysis system. An example of the interface is shown in Figure 15. The display both provided a means of evaluating the quality of the T–LAPS analysis and provided a vehicle for demonstrating the capabilities of the T–LAPS analysis system. Additionally, the display in Lexington provided a means of monitoring the weather situation in Orlando, Florida. Due to considerations which are discussed below, different displays were provided at different locations.

6.1. Design Considerations

Three considerations significantly affected the design of the display system. The first was the real–time nature of the display. The second was the limited time in which to construct the display system. The third was the requirement for remote displays combined with a limitation of 9600–baud bandwidth.

The first two considerations resulted in the decision to use the same display protocol as used for the TDWR Geographic Situation Display (GSD). This display protocol will be described in section 6.3. The third consideration resulted in the decision to provide limited displays at remote locations. This, combined with differing requirements at different remote locations, resulted in three different displays. The data provided to these displays will be discussed in the following sections.

For T–LAPS '92, no effort was made to display the results of the T–LAPS 10 km processing because the 3–D wind products from the T–LAPS 2 km analysis were more timely and of finer resolution than those of the T–LAPS 10 km analysis.

6.2. Display Product Server

The display product server provided the interface between the T–LAPS analysis system and the display software. It was responsible for preparing the products for display and for transmitting the product information to the various displays. Figure 16 depicts the relationship between the display product server, the T–LAPS 2 km winds analysis, and the various displays.

6.2.1. Initialization

The display product processor was started during initialization of the T–LAPS analysis system. This is described in section 5.1. When the display product server was first started, run–time options were used to initialize access to the proper directories, initialize run–time parameters, and establish the communication channels. Parameters which were likely to change due to system reconfiguration were defined as run–time options. These included the name of the signal file to be created by the driver, the location of the data directories, and polling parameters. Once the display product server had been initialized, it began polling for the file that was used to signal the availability of data for display.

6.2.2. Data Ingest

Upon receipt of the signal file, the display product server searched in the specified directories for the 3–D background, 3–D wind, and 3–D radar files. In general, the most recent files with the appropriate suffixes were used. Although two 3–D radar files were used by the T–LAPS analysis

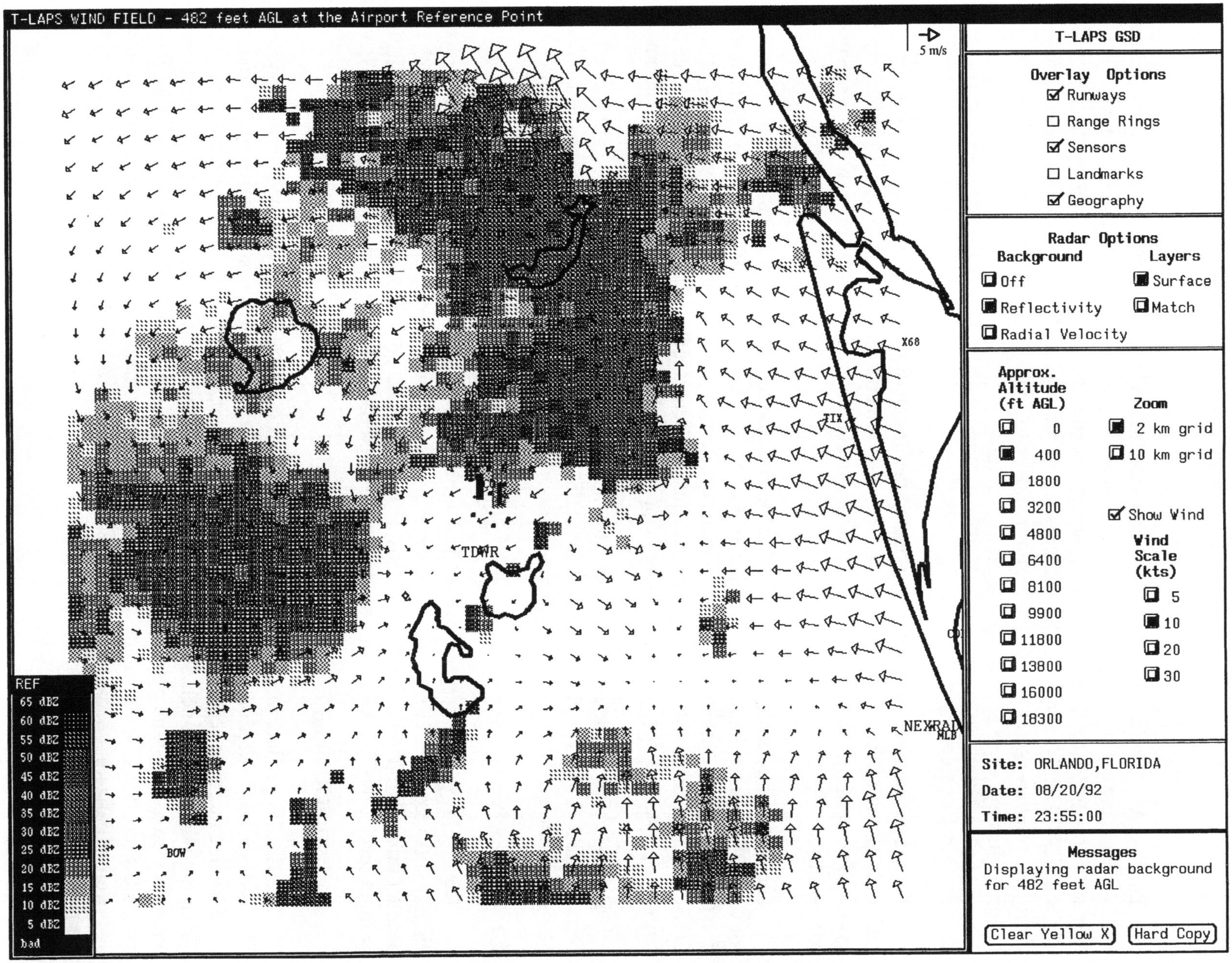

Figure 15. Format of T–LAPS display used at FL–2C.

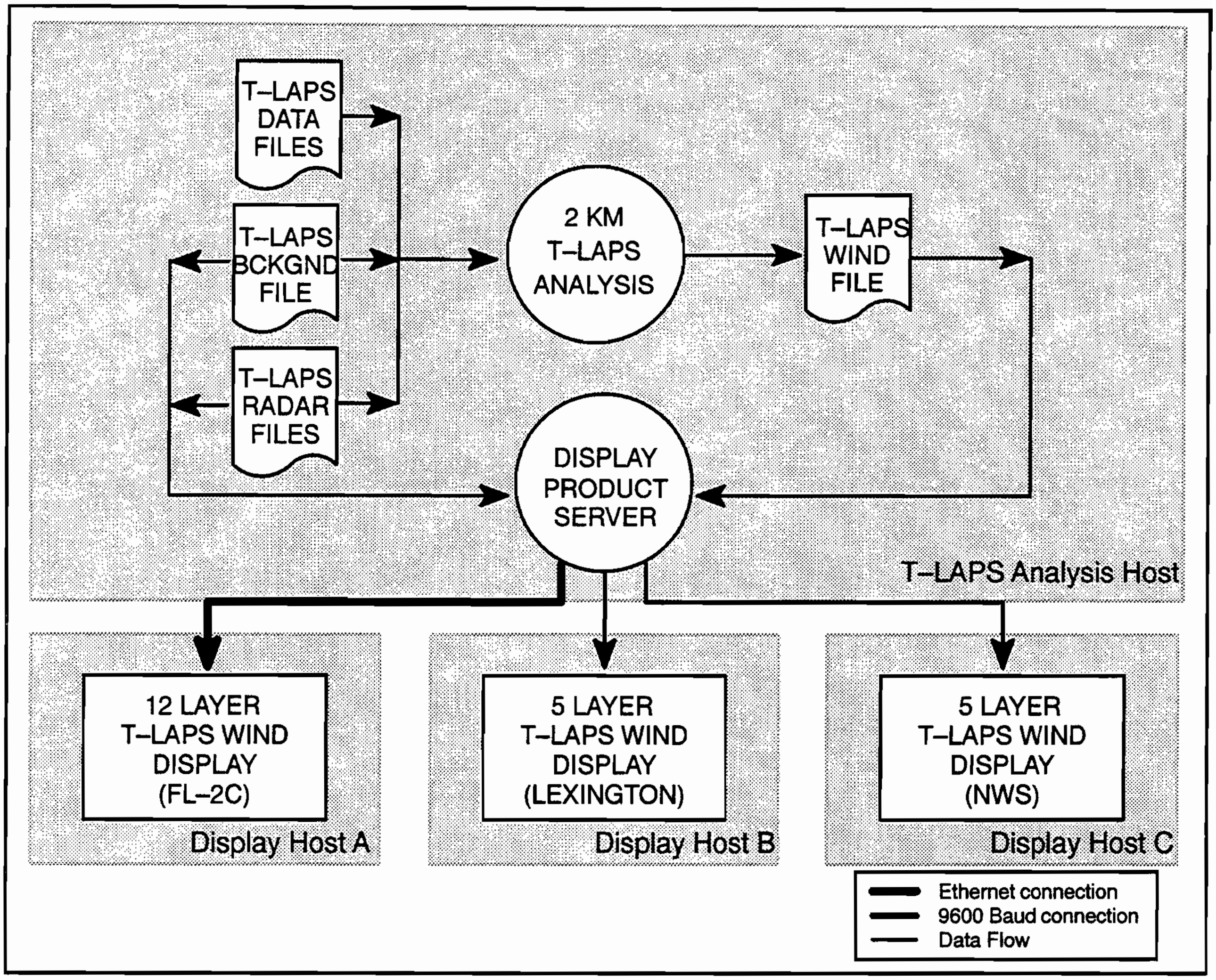

Figure 16. Display product server processing.

module, only one radar file was transmitted to the display. Initially, only the TDWR file was transmitted. Later, however, the program was changed to transmit the NEXRAD file when the TDWR file was unavailable or empty. Processing continued in the absence of any radar files. However, the background and wind files were required for processing to continue.

6.2.3. Processing

Once the appropriate files were identified, the processing proceeded in three steps: obtain altitude information, perform data conversion, and transmit the information. The last topic is the subject of the next section.

As mentioned in previous sections, the native LAPS vertical coordinates are millibars. However, for display purposes, it was desirable to display the vertical coordinates as feet above ground level. Thus, it was necessary to obtain information regarding the association between the two. This was accomplished by taking advantage of the height field in the background file. The value of this field was obtained for the horizontal grid point aligned with the airport reference point for each layer of data in the file. Thus, each layer was effectively considered to be at a uniform height in feet above ground level. This was a sufficient approximation for purposes of the display. It should be noted that although the layers were evenly spaced in pressure coordinates, they were not evenly spaced in altitude coordinates. This was judged to be unimportant for the purposes of the T–LAPS '92 demonstration.

Once the altitude coordinates were obtained, the data was then converted from the native LAPS format to a format recognizable by pre–existing Lincoln Laboratory software. Due to constraints imposed by the write package for the data format, it was necessary to first create files on disk, which could subsequently be read and transmitted. Since it was necessary to support three different categories of display, each requiring different layers of data, one option was to create separate files based on the requirements of each display. However, it was decided that this would be expensive in terms of both disk resources and disk input/output (I/O). So the decision was made to produce a single file which would be filtered during transmission.

During the conversion process the number of wind vectors was reduced by a factor of four by converting only every other row and column of data. This had the dual benefit of producing displays which were easier to read and reducing the bandwidth requirements. Since the radar display would suffer from a similar reduction in data, such a reduction was not performed for those data.

6.2.4. Product Serving

The last responsibility of the display product server was to transmit the prepared data to the displays. As mentioned before, there were three categories of displays: primary (FL–2C, Kissimmee, FL), demonstration (NWS, Melbourne, FL), and monitoring (Lexington, MA). The latter two were remote displays and thus could display only 5 of the 12 available layers of 3–D winds and only one of the 12 available layers of 3–D radar.

The primary processing task for file transmission involved the creation of three valid display files from each single file residing on the disk. The radar file was packaged and transmitted first, followed by the wind file. Each file was processed one record at a time. The header record was read once. For each display category, the header was edited based on the layers to be displayed, compressed, and transmitted to the appropriate client. Each subsequent record in the file was read once and, for each display category, compressed and transmitted to the appropriate client, as needed. Once file transmission was completed, the scratch files were removed and the display product server waited for the appearance of the next signal file.

6.2.5. Termination

The display product server was terminated by a SIGTERM signal. Upon receipt of the termination signal, a flag was set to indicate that the process should terminate upon completion of any active processing. Due to an incomplete implementation of the signal handler, the display product server occasionally failed to terminate. The problem has since been fixed.

6.3. Product Display

The product display software consisted of communications and monitoring programs as well as the user interface. The configuration of the display system was modeled, to a large degree, on the TDWR GSD. Figure 17 depicts the suite of programs associated with the display.

6.3.1. Initialization

During initialization a single script was used to start a suite of programs associated with the display. During real–time operations, this script was primarily invoked upon log in to a special display

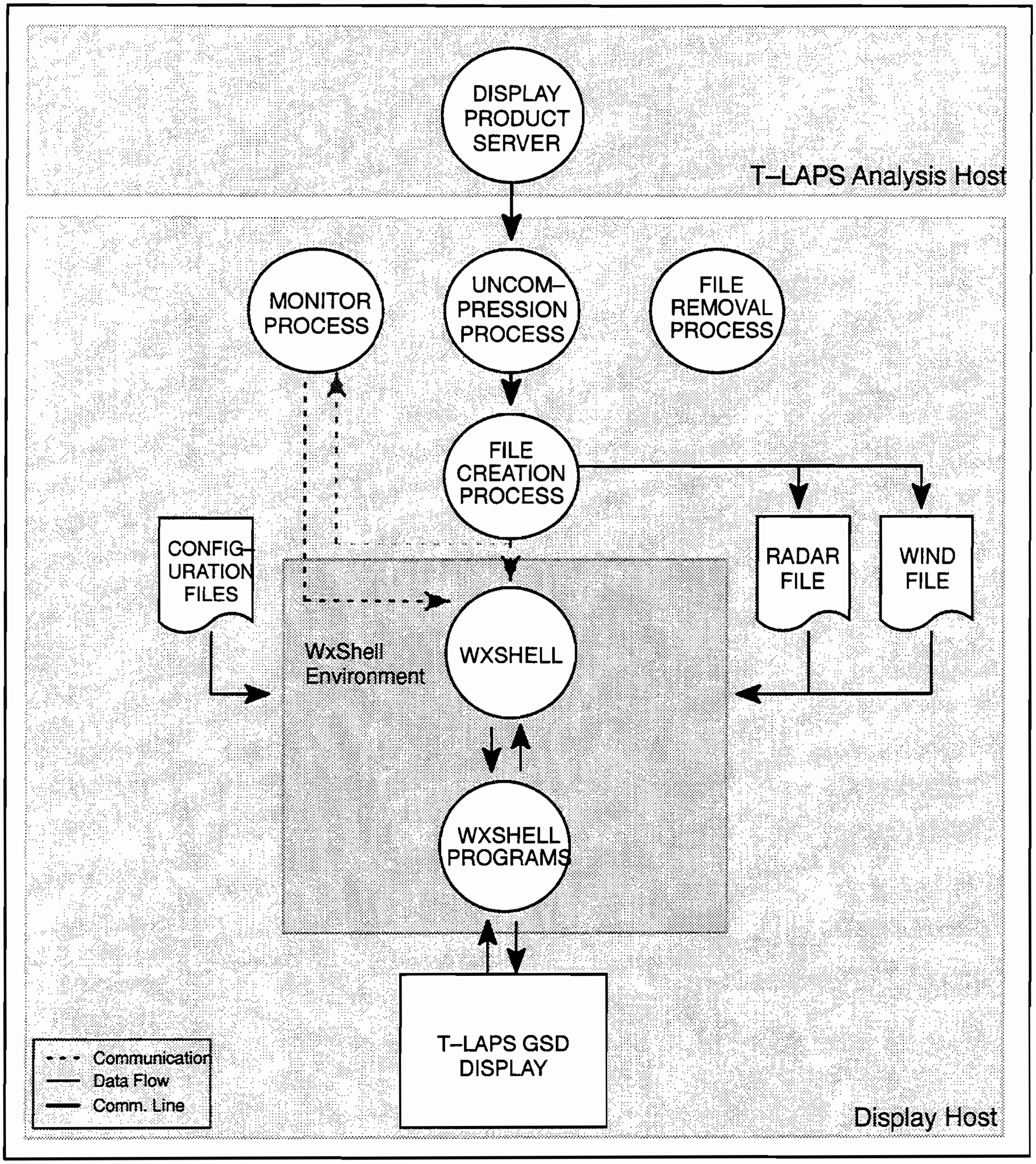

Figure 17. Real–time product display.

account. It could also be invoked from the command line or from a root menu. In Lexington, this script was invoked automatically each morning to reinitialize the display. During initialization, the user interface was displayed with a red X overlaid on the display area. This X was removed from the display once the first 3–D winds display file was received.

One program started during initialization was responsible for creating the directory structure to be used as working directories. Another program was responsible for establishing a communications channel to the display product server, described above. The suite of programs as well as the programs themselves, while generally the same for all locations, were to some degree site specific. For example, the program that created a directory structure was configured to archive the display files directly into the database for the display located in Lexington. Accordingly, no program was necessary in Lexington to delete old display files, whereas such a program was necessary at other locations.

6.3.2. Data Ingest

Data was ingested into the display in three steps. The actual display software read in a data file. A separate process read information from a communications channel and generated a display file. This intermediate process was, however, unable to read compressed data. Another process was required which read from the communications channel to the display product server and generated a separate channel of uncompressed data which could be read by the second process.

6.3.3. Display Processing

As mentioned previously, the display system was built using the same technology as the TDWR GSD. The base for this system was an interpreter called Weather Shell ("WxShell" in Figure 17), which provides facilities for interactive displays and real–time communications. Additionally, using Weather Shell provided support for multiple Sun architectures and windowing systems. Although much of the display capabilities already existed in the form of Weather Shell and its clients, additional software was developed to handle the specific requirements of the T–LAPS display.

The body of the software that controlled direct interaction with the display remained invariant from location to location. The differences among locations were supported by providing different input panels for each location, which allowed only a selection of options valid for that location, and by providing location–specific configuration files.

At all locations, the user was able to select between radial velocity and reflectivity displays as well as among horizontal wind fields at selected altitudes. The user also was able to activate or deactivate displays of sensors, runways, geography, landmarks, and range rings, adjust the scale of the wind vectors, and select between two display scales. Additional options were available on selected displays.

6.3.4. Display Termination

The display was terminated by positioning the cursor over the display window and typing 'q'. This resulted in the termination of the display as well as all associated processes. Additionally, in Lexington, a termination command was sent automatically each morning which had the same result.

7. ARCHIVING

This section discusses the archiving procedures used for the T–LAPS 1992 real–time system. The primary purpose of archiving was to support post–real–time evaluation of the T–LAPS wind analyses. To attain this objective, raw and intermediate data were stored as well as the results of the analysis. In this context, raw data refers to data ingested into the T–LAPS system from external sources, as discussed in section 3. Intermediate data refers to data processed for input to the analysis packages, as discussed in sections 3. and 5. In some cases, archiving procedures were already in place for some of the raw data of interest. When archiving procedures were in existence, they were not duplicated.

There were two categories of archiving procedures. The first involved storing particular files from the hard disk onto a storage medium. The second involved the acquisition of data files via file transfer. Both categories of archiving will be discussed below. In some cases, this storage of information was redundant. The redundancy was introduced when data were needed in Lexington in a more timely fashion than would otherwise be available.

7.1. Storage of T–LAPS Data at FL–2C

The 1992 T–LAPS system ran for up to 14 hours per day, 7 days per week. Two Sun SPARCstations at FL–2C were used for the bulk of the real–time processing. Both of these workstations were dedicated to T–LAPS during the summer demonstration. Each of them had a 420 Mbyte disk, of which approximately 280 Mbytes was free for use by T–LAPS. The workstations were networked via the NFS, so files on the two disks were accessible to processes running on either machine.

The T–LAPS processes expected files to be in specific directories, referred to hereafter as working directories. As discussed in section 5.5., in order to minimize the number of files resident in any given T–LAPS working directory at one time, files were moved to an archive directory periodically during T–LAPS processing as well as at the end of T–LAPS processing. An estimate of the disk space required to store T–LAPS related files during each one–hour interval is shown in Table 1. The values in the table represent the size of the files at the beginning of the demonstration period. By the end of the demonstration period, the size of the wind analysis files had been reduced by roughly one third. Although compression was desirable, the LAPS balance output files, radar volume files, and T–LAPS wind analysis output files, which occupied the majority of the disk space, were essentially incompressible by the Unix `compress` utility. Because of this and the high value placed on CPU resources, compression was not performed during operations.

The hourly total of almost 12 Mbytes per hour adds up to about 160 Mbytes over a 14–hour day. Thus, only one day of real–time data could be safely stored on one workstation's disk at any one time. In order that site personnel were not forced to back up the archive directories after every single day of operations, archive directories were created on both of the dedicated SPARCstations (named TAPS1 and TAPS2), as shown in Figure 18. Symbolic links (indicated in the diagram by arrows) were used to facilitate switching between the archive directories by the T–LAPS real–time system driver.

Table 1.
Approximate T–LAPS Space Requirements for One Hour of Operations

Data Files	Dimensions row x col x layer x prod	Size (Kbytes)	Files per Hour	Kbytes per Hour
2 km wind analysis	61x61x13x3.15	616	12	7392
10 km wind analysis	19x19x13x3.15	128	2	256
10 km balanced winds	19x19x21x4	263	2	526
Radar volume	N/A	11	2x12	264
2 km radar tilt files	N/A	16	2x2x12	768
10 km LLWAS	N/A	1.5	2	3
2 km LLWAS	N/A	1.5	12	18
SAO/AWOS	N/A	5	12	60
ACARS	N/A	5	6	30
MAPS	6x6x21x5	115	2x1/3	77
10 km background	19x19x21x3	197	2	394
2 km background	61x61x21x3	946	2	1892
			TOTAL	11680

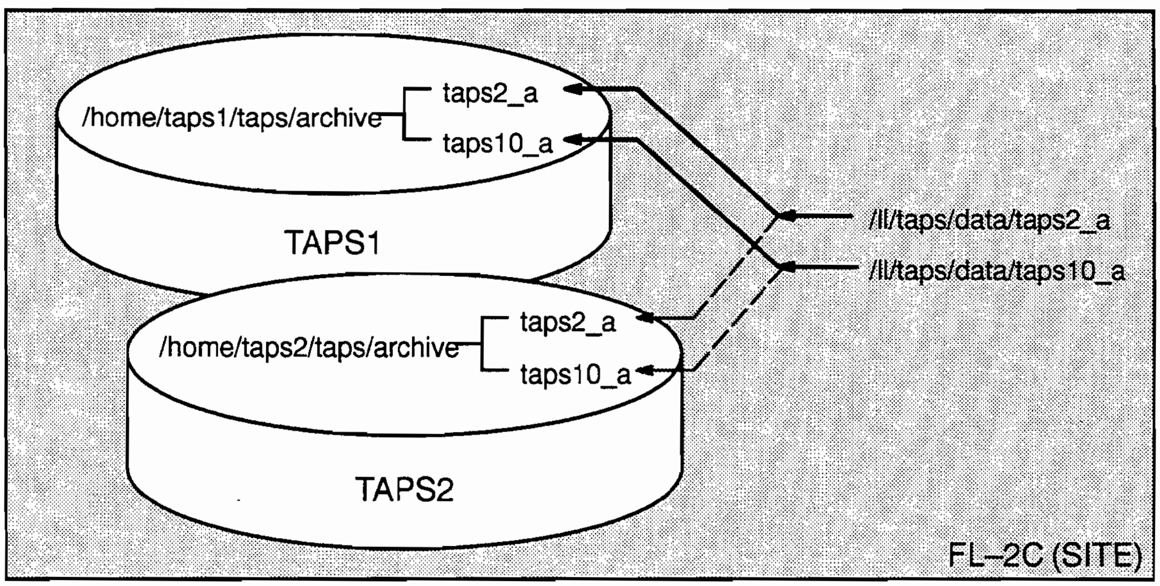

Figure 18. Relationship between physical disks and working space.

7.2. Archiving to Tape at FL–2C

When the system was operating smoothly, archiving of the data generated during real–time operations to tape was performed following termination of T–LAPS processing each day. Information relevant to this archive was logged to a file. This file was mailed to the archive administrator once the archiving to tape was completed. A lock file in each archive directory indicated whether the data in that directory had been archived to tape. Any directory that was not previously archived to tape was archived at the end of the day. The lock files also were checked prior to start of T–LAPS operations. If the lock file indicated that the data were backed up to tape, the contents of the archive directory were removed so that the directory could be reused. If neither archive directory had been stored to tape, an error message was displayed. The operator was then required to archive the data to tape before processing could proceed.

Initially, the use of File Transfer Protocol (FTP) over a pre–existing Internet Protocol (IP) network link was considered to allow archiving onto storage media to be performed in Lexington. However, this was rejected due to the large volume of data that must be transferred and the relatively low bandwidth and unreliability of that link. The archiving onto magnetic media was therefore performed at site. It was desirable to select a storage medium that would not require supervision during archiving. Due to the large quantity of data generated each day, 8 mm tape was selected as the archive storage medium. Approximately one week of real–time data was stored on a single tape, with each day of data stored as a separate file system. Tapes were mailed to Lexington approximately every week.

7.3. Archiving via File Transfer

7.3.1. From External Agencies

All MAPS and surface observation files for the previous day were transferred from their original sources (FSL and NASA, respectively) via separate `cron` jobs run at Lexington each night. As discussed in section 3., transfer of MAPS data used Kermit to access the modem, and transfer of NASA data used `tip`. These data transfers were deliberately redundant with the transfers performed by the real–time system. They provided a means of getting more complete data sets and retrieving data that may have become available too late for use in the real–time system. The latter was particularly pertinent in the case of MAPS data. Performing the archiving of these data sets in this manner also shifted some responsibility for archiving off of the real–time system driver.

Since the underlying software was largely the same as that described in section 3., it exhibited some of the same problems. In addition, since the transfers occurred at night, there was a lapse between the time a problem occurred and the time it was noticed. However, there were few failures and the reliability of these transfers was improved by the end of the demonstration period.

7.3.2. From the T–LAPS Real–Time System Disks

The radar tilt files for the previous day were transferred from the site to Lexington via a `cron` job run from Lexington each night. These tilt files were located in the archive directories at the site and were transferred via remote UNIX commands over the dedicated line to FL–2C. The log files from the tape archive were also transferred at this time.

On the whole, this worked reliably. There were a few instances where tilt files were empty. Since the same files were archived on tape and also could be retrieved the following day, this did not present a problem.

7.3.3. From Real–Time Data Feeds

Finally, the display files, which were transferred to Lexington for the real–time display, were archived as they arrived. The display was started in Lexington each morning by a `cron` job and ran continuously throughout the day. This worked reliably. On one occasion the space on the disk was exhausted and some data were lost. Once this problem was addressed, there were no further problems.

7.3.4. Archiving by Others

There has been Lincoln Laboratory interest in ACARS data that pre–dates the T–LAPS project. There was already a pre–existing facility for downloading, via dial–up, ACARS data from ARINC to Lexington. Although it was not originally designed as a T–LAPS backup source of data, this facility was functionally similar to the redundant downloads of MAPS and surface station data discussed in section 7.3.1.

Both TDWR and NEXRAD data were archived at their point of origin. TDWR data were archived at the FL–2C site by site personnel, while NEXRAD data was archived in Melbourne by NWS personnel. Due to the size of the radar data and the existence of established group procedures for handling radar data, the data was not stored in the actual T–LAPS database. However, the stored radar data was still very much a part of the overall T–LAPS dataset. New versions of the resampled radar files could be regenerated, if necessary, by running a playback version of the resampler on the stored radar data. The resulting tilt files would then become a part of the formal T–LAPS database.

7.4. Organization of the T–LAPS Database

During the demonstration period, the data collected via the various archive procedures was collected into a database on a 1.2 Gigabyte disk. Once a complete day's worth of data had been accumulated, the entire day's data was stored on 8 mm tape. `Cron` jobs ran each morning to process and organize the data transferred overnight and during the previous day. When a tape arrived from site, the data for each day were processed and stored in the proper directory. The archive for a day was considered complete once all of the data for that day had been processed, organized, and logged. The collection of data into the database is depicted in Figure 19.

Once one day's worth of data were collected, there were no significant problems with organizing it and creating a tape archive. For most cases, the data were stored in the database without additional processing, other than compression. The one exception was the MAPS files obtained from FSL. In this case it was necessary to convert from VAX binary format to Sun binary format. This procedure was performed when the files were stored into the database.

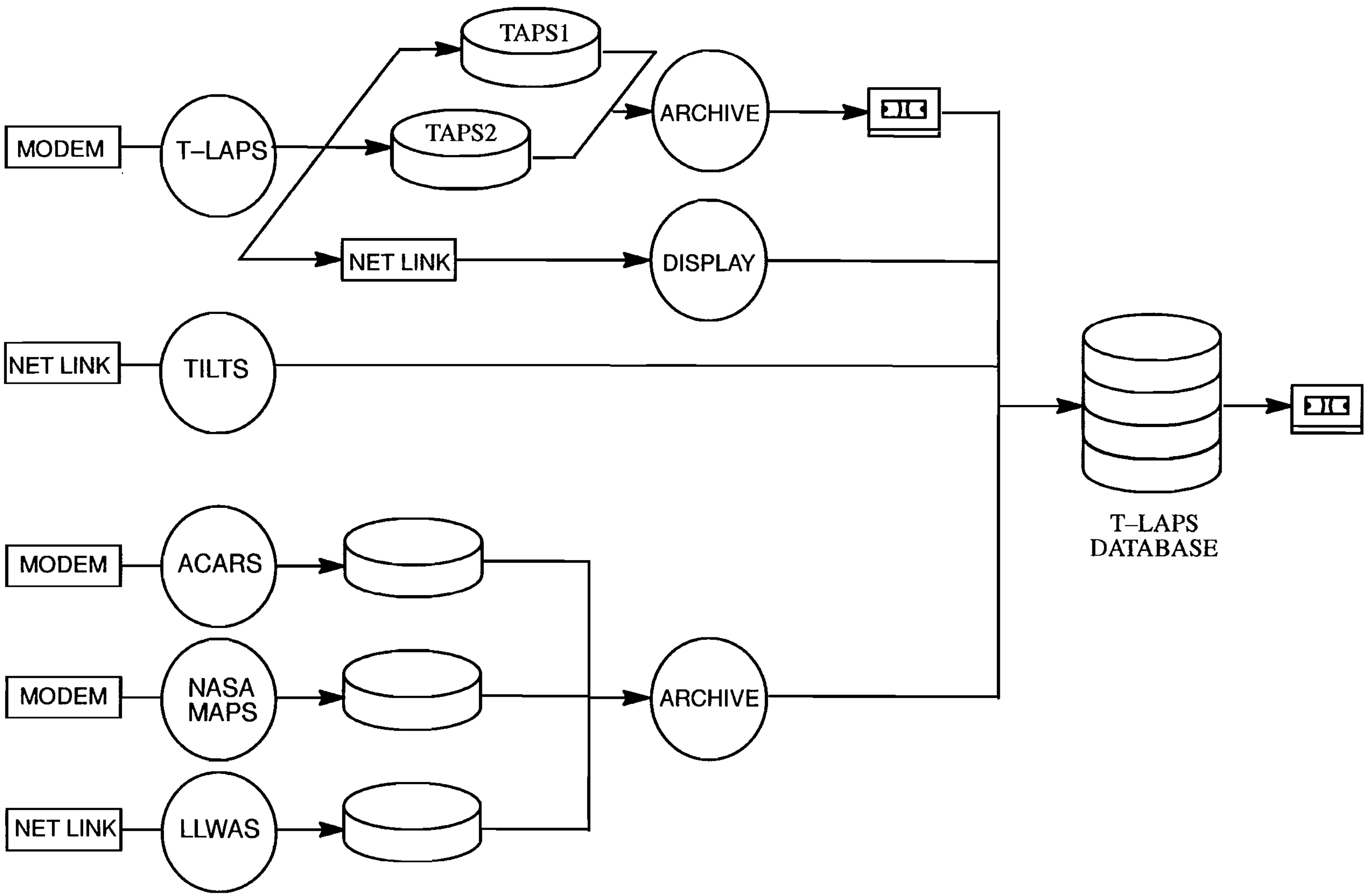

Figure 19. Data flow into the T–LAPS database.

8. CONCLUSION

The T–LAPS system was installed at the FL–2C site in late July 1993. Testing the system and fixing problems occupied the first two weeks of August. The T–LAPS demonstration officially began August 14, which was only a few days after the NEXRAD data became available. After some subtle problems that affected the quality of the output were fixed, the system started producing high–quality results on August 22. The system continued to run without serious incident until September 25, when the system was shut down.

Overall, the reliability of the system was very good. After the initial testing period was completed, virtually all the ongoing problems were related to acquiring data from outside organizations. The system was designed to be robust in the face of missing data, so the system ran without a single outright failure during the entire demonstration period.

The T–LAPS '92 demonstration was the first time that TDWR and NEXRAD data had been used simultaneously in a single system. The T–LAPS '92 demonstration also marked the first demonstration of a prototype ITWS product using a wide variety of data sources. The lessons learned will be extremely valuable as the group fields more ITWS algorithms in the future.

Most importantly, the final result of the system was a high–quality dataset that could be used for later playback and analysis. The redundant downloading of data directly to Lexington allowed filling in some stretches of missing data. The dataset has allowed both Lincoln Laboratory and FSL to refine existing algorithms and develop new algorithms.

GLOSSARY

2–D	Two dimensional
3–D	Three dimensional
ACARS	Aircraft Communications Addressing and Reporting System
AGFS	Aviation Gridded Forecast System
ARINC	Aeronautical Radio, Inc.
ASCII	American Standard Code for Information Interchange
AWOS	Automated Weather Observing System
BUFR	Binary Universal Form for Representation of Meteorological Data
FAA	Federal Aviation Administration
FSL	Forecast Systems Laboratory
FTP	File Transfer Protocol
GSD	Geographic Situation Display
IP	Internet Protocol
ITWS	Integrated Terminal Weather System
LAPS	Local Analysis and Prediction System
LLWAS	Low Level Wind Shear Alert System
MAPS	Mesoscale Analysis and Prediction System
MDCRS	Meteorological Data Collection and Reporting System
NASA	National Aeronautics and Space Administration
NEXRAD	Next Generation Weather Radar (WSR–88D)
NFS	Network File System
NOAA	National Oceanic and Atmospheric Administration
NWS	National Weather Service
RCS	Revision Control System
S/N	Signal to Noise ratio
SAO	Surface Aviation Observation
T–LAPS	Terminal area–Local Analysis and Prediction System
TATCA	Terminal Air Traffic Control Automation
TDWR	Terminal Doppler Weather Radar

REFERENCES

Albers, S. C., 1992: The LAPS wind analysis. *Fourth Workshop on Operational Meteorology*, Whistler, B.C., Canada.

Andrews, J, and J. Welch, 1989: The Challenge of Terminal Air Traffic Control Automation. Proceeding, *34th Air Traffic Control Association Conference.*

Barnes, S., 1964: A technique for maximizing details in numerical weather map analysis. *J. Appl. Met.*, 3, No 4.

Benjamin, S. G., *et al.*, 1991: An isentropic three–hourly data assimilation system using ACARS aircraft observations. *Monthly Weather Review,* Vol. 119, No 4.

Cole, R.A., F.W. Wilson, J.A. McGinley, and S.C. Albers, 1993: ITWS Terminal Gridded Analysis. Preprints, *Fifth Conference on Aviation Weather Systems*, Vienna, Virginia, American Meteorological Society.

Crum, T.D. and R.L. Alberty, 1993: The WSR–88D and the WSR–88D operational support facility. *Bull. Amer. Meteor. Soc.,* Vol. 74, No. 4, 645–653.

Dey, C., R.C. Martin, and R. Peterson, 1991: The Meteorological Data Collection and Reporting System (MDCRS): System Overview and Benefits. Postprints, *NWS Aviation Weather Workshop*, Kansas City. NOAA Technical Memorandum NWS CR–102.

Ducot, E., 1993: Designing and engineering the Integrated Terminal Weather System. Preprints, *Fifth Conference on Aviation Weather Systems*, Vienna, Virginia, American Meteorological Society.

Evans, J.E., 1991: Integrated Terminal Weather System (ITWS). Preprints, *Fourth International Conference on the Aviation Weather System,* Paris.

Kraus, M.J., 1993: The Forecast Systems Laboratory's role in the FAA's Aviation Weather Development Program. Preprints, *Fifth Conference on Aviation Weather Systems*, Vienna, Virginia, American Meteorological Society.

Martin, R.C., M.W. Wolfson, and R.G. Hallowell, 1993: MDCRS: Aircraft Observations Collections and Uses. Preprints, *Fifth Conference on Aviation Weather Systems*, Vienna, Virginia, American Meteorological Society.

McGinley, J., S. Albers, P. Stamus, 1991: Validation of a composite convective index as defined by a real–time local analysis system. *Wea. Forecast.*, 6, 337–356.

Merritt, M.W., D. Klingle–Wilson, and S.D. Campbell, 1989: Wind shear detection with pencil–beam radars. *Lincoln Laboratory Journal,* Vol. 2, No. 3.

NOAA–National Weather Service, 1979: Aviation Automated Weather Observation System (AV–AWOS). Report FAA–RD–79–63, National Technical Information Service, Springfield, Virginia.

Sankey D.A., and A.L. Hansen, 1993: FAA's Work in Improving Aviation Weather. Preprints, *9th International Conference on Interactive Information and Processing Systems.* Anaheim, CA.

Schlatter, T.W., 1991: A Rapid Update Analysis and Prediction System at NMC for Aviation Forecasting. Postprints, *NWS Aviation Weather Workshop*, Kansas City. NOAA Technical Memorandum NWS CR–102.

Sherretz, L.A., 1991: Developing the Aviation Gridded Forecast System. *Fourth International Conference on the Aviation Weather System,* Paris.

Wilson, F.W. and R.H. Gramzow, 1991: The redesigned Low Level Wind Shear Alert System. *Fourth International Conference on the Aviation Weather System,* Paris.