TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA/TX-87/283-3F		

4. Title and Subtitle	5. Report Date
Modeling and Measurement of Vehicle Emissions From Roadways	August 1987
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
J. A. Bullin, M. W. Hlavinka, J. J. Korpics, J. H. Schroeder, R. D. Moe, G. R. Donaldson	Research Report 283-3F

9. Performing Organization Name and Address	10. Work Unit No.
Texas Transportation Institute The Texas A&M University System College Station, Texas 77843	11. Contract or Grant No. Study No. 2-8-80-283
	13. Type of Report and Period Covered Final - September 1979 August 1987
12. Sponsoring Agency Name and Address Texas State Department of Highways and Public Transportation; Transportation Planning Division P. O. Box 5051 Austin, Texas 78763-5051	14. Sponsoring Agency Code

15. Supplementary Notes

Research performed in cooperation with DOT, FHWA.
Research Study Title: Vehicle Emissions from Roadways

16. Abstract

Under project 283, both experimental and model development work in air pollution research near roadways was considered. First, the original TXLINE model was modified so that it was suitable for use in modeling finite line sources. This modification enables the model to be used in modeling pollutant dispersion on curved roads and other types of scenarios for which an infinite line source model would not be applicable. This modification also allows the model to be used in predicting pollutant concentrations upwind of a roadway.

The next research area considered was the revision of the original Texas intersection model, TEXIN. The original model had several limitations which inhibited its use in a large number of cases. With the revised version, many of these limitations are no longer present. T-intersections are specifically treated by appropriately assigning internal variables. Improved emission factor estimates are obtained with MOBILE3. MOBILE3 allows the user enhanced flexibility in describing the vehicle distribution along with anti-tampering options and inspection/maintenance programs. The user may choose either the CMA Operations and Design algorithm or the CMA Planning Procedure to analyze traffic flow at the intersection. The model exhibits very good accuracy for areas not surrounded by tall buildings.

Finally, the collection of a large experimental data base was performed under the project. Data on meteorology, traffic, and pollutant concentrations were obtained in Houston, Texas. These data were (CONTINUED ON THE BACK OF THIS PAGE)

17. Key Words	18. Distribution Statement
Carbon Monoxide Concentrations, Dispersion Modeling, Vehicle Emissions, Mass Balance Technique, Computer Data Acquisition, Intersection Modeling, CALINE3, TEXIN2, TXLINE-2.	No restrictions. This document is availabie to the public through the National Technical Information Service 5285 Port Royal Road Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	362	

Form DOT F 1700.7 (8-69)

used in the mass balance technique to estimate carbon monoxide emission factors which were then compared to the simulated factors obtained by the EPA model MOBILE3. In general, the MOBILE3 simulations yielded lower emission rates that those calculated by the mass balance. The primary purpose of the data base is the improvement and development of future roadway air quality models. In order to facilitate this process, the data base is available to the public on standard nine-track tape. Fifteen minute and hourly averages are printed in Appendix K, which is bound in a separate volume of this report.

Modeling and Measurement of Vehicle Emissions from Roadways

by

Jerry A. Bullin
Michael W. Hlavinka
John J. Korpics
James H. Schroeder
Roderick D. Moe
Guy R. Donaldson

Texas Transportation Institute
Texas A&M University System
College Station, Texas 77843

Sponsored by

Texas State Department of Highways and Public Transportation
in cooperation with
U. S. Department of Transportation
Federal Highway Administration

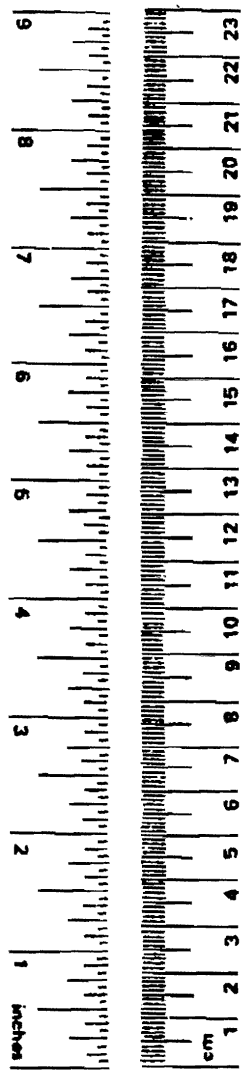
Research Report 283-3F
Research Study No. 2-8-80-283
Vehicle Emissions from Roadways

July 1987

METRIC CONVERSION FACTORS

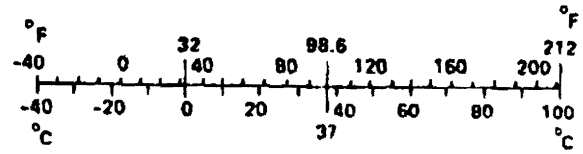
Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.

Table of Contents

*NOTE: Appendix K will be a part of this report but will be under a separate cover due to its length.

List of Figures

List of Tables

Implementation

Air quality data have been collected at an at-grade research site in Houston, Texas. The data have been arranged in a manner suitable for future model development or validation. Two revised pollutant models have been released under this project. The first of these is a finite line source model while the second is a composite, intersection model. This work should enable the preparation of more accurate environmental impact statements.

Disclaimer

The contents of this report reflect the views of the authors who are responsible for the facts and the data presented herein. The contents do not necessarily reflect the official views or policies of the Federal Highway Administration, nor does this report constitute a standard, specification, or regulation.

Acknowledgements

The authors wish to recognize the contributions made by several individuals during the project. Mr. Roger Wayson of the Texas State Department of Highways and Public Transportation was helpful with the research site preparation and the tracer gas experiments. Mr. John Hogue of the Department of Chemical Engineering at Texas A&M University was extremely helpful with many of the electronic problems that occurred in the experimental phase. The devotion of his expertise in software development is also greatly appreciated. Thanks go to Drs. F. A. Godshall and A. R. McFarland of the Department of Environmental Engineering at Texas A&M University for their technical suggestions in the project. We appreciate the data and other information supplied by Mr. Paul Benson of the California Department of Transportation for use in the verification of TEXIN2. Comments received from Dr. Amulakh Parikh of the New Jersey Department of Transportation were used in the TEXIN2 traffic analyses. Thanks go to Laura Lapaglia for her diligent work in assembling the final draft of this report. Special thanks are extended to Mrs. Laura Hlavinka for her contribution in preparing this manuscript and assistance in the tracer gas experiments. As always, the staff support of the Texas Transportation Institute and the Chemical Engineering Department at Texas A&M University was greatly appreciated.

Summary

Under Project 283, both experimental and model development work in air pollution research near roadways was considered. First, the original TXLINE model was modified so that it was suitable for use in modeling finite line sources. This modification enables the model to be used in modeling pollutant dispersion on curved roads and other types of scenarios for which an infinite line source model would not be applicable. This modification also allows the model to be used in predicting pollutant concentrations upwind of a roadway.

The next research area considered was the revision of the original Texas intersection model, TEXIN. The original model had several limitations which inhibited its use in a large number of cases. With the revised version, many of these limitations are no longer present. T-intersections are specifically treated by appropriately assigning internal variables. Improved emission factor estimates are obtained with MOBILE3. MOBILE3 allows the user enhanced flexibility in describing the vehicle distribution along with anti-tampering options and inspection/maintenance programs. The user may choose either the CMA Operations and Design algorithm or the CMA Planning Procedure to analyze traffic flow at the intersection. The model exhibits very good accuracy for areas not surrounded by tall buildings.

Finally, the collection of a large experimental data base was performed under the project. Data on meteorology, traffic, and pollutant concentrations were obtained in Houston, Texas. These data were used in the mass balance technique to estimate carbon monoxide emission factors which were then compared to the simulated emission factors obtained by the EPA model MOBILE3. In general, the MOBILE3 simulations yielded lower emission rates than those calculated by the mass balance. The primary purpose of the data base is the improvement and development of future roadway air quality models. In order to facilitate this process, the data base is available to the public on standard nine-track tape. Fifteen minute and hourly averages are printed in Appendix K, which is bound in a separate volume of this report.

Chapter 1

Introduction

The National Environmental Policy Act of 1969 dictates that environmental impact statements must be submitted to the Federal Highway Administration before the start of any major roadway construction project.[1] This report must be reviewed by governing agencies such as the Federal Highway Administration, the Environmental Protection Agency, the State Department of Highways and Public Transportation, and the Texas Air Control Board. The environmental impact statement must include predictions on pollutant concentrations in the mesoscale and microscale areas of the roadway for approximately 20 years after the estimated time of completion of the construction project. The report must consider at least two cases: (1) the proposed project is undertaken and (2) the proposed project is not undertaken. The Federal Aid Highway Act of 1970 establishes procedures to assure that all possible adverse economic, social, and environmental effects have been considered while developing the proposed project.

The primary pollutants arising from internal combustion engines are: (1) carbon monoxide, (2) oxides of nitrogen, (3) hydrocarbons, (4) lead, and (5) particulate matter. Secondary pollutants, such as ozone, are formed in the presence of ultraviolet light and the primary pollutants. The principal factors that affect the concentration of these pollutants at a downwind receptor include the distance from the pollution source, the source strength, the mixing height, the wind speed, the wind direction, and the associated atmospheric and induced turbulence.

Several models (TXLINE,[2] CALINE4,[3] and HIWAY-2[4]) have been proposed to aid the transportation engineer in developing environmental impact statements. Composite models have also been constructed in order to predict pollutant concentrations near intersections (TEXIN,[5] IMM,[6] and MICRO[7,8]). The basic logic used in modeling dispersion near roadways includes two major steps. The first of these involves the estimation of emissions due to the motor vehicles. Secondly, atmospheric dispersion principles are applied along with the prevailing meteorology to estimate the pollutant concentrations at a given receptor. Composite intersection models also include many traffic algorithms in order to characterize the traffic flow.

The modeling of pollutant dispersion near roadways is a complicated process. The preparation of highway dispersion models requires many hours of experimental data. Pollutant concentrations at various receptor locations must be recorded along with the associated meteorology and traffic data. These data bases are often used to derive empirical relations in the dispersion models where the theory is insufficient or the mathematics is too complex. Furthermore, these data are used in statistical analyses of the performance of the model.

Objectives

The overall objectives of project 283 are to assist in preparation of more accurate environmental impact statements and to enable a better estimation of the effects of air pollution in sensitive areas. These objectives are accomplished through the following major tasks:

(1) Revision of the TXLINE model so that it may be used for modeling finite line sources and predict pollutant concentrations upwind of a roadway.

(2) Revision of the TEXIN model by adding new traffic estimation algorithms and new emissions calculational procedures.

(3) Collection of a data base including all the required parameters suitable for use in improving and evaluating model performance in the future.

(4) Estimation of vehicular emissions from the experimental data and comparison to MOBILE3 simulations.

Chapter 2

Literature Review

The task of developing models to estimate the pollutant dispersion near roadways has traditionally been accomplished by first estimating the source strength by invoking an emissions model for the vehicle scenario in the region. Subsequently, the dispersion of the pollutant in the atmosphere is normally modeled by a Gaussian dispersion or gradient transport model. Finally, the model is compared to several existing data bases to determine its accuracy. Therefore, this chapter is divided into the above topics.

I. Determination of Source Strength

Every atmospheric dispersion model requires an estimation of the source strength. This is quite easily done for stationary sources, but may be extremely difficult for motor vehicles. The Environmental Protection Agency (EPA) has administered several exhaust emission studies. These studies have resulted in several emission models that are described below.

A. AP-42

The EPA has developed several standard driving sequences to represent urban emissions. Data collected using the Federal Test Procedures (FTP) and Surveillance Driving Sequences (SDS) have been combined with assembly line test data, prototype models, test of in use vehicles, tampering surveys, and technical judgement to form the basis for the existing and projected mobile source emission factors. This study is presented in the EPA document, *Compilation of Air Pollutant Emission Factors: Highway Mobile Sources* (AP-42).[9]

B. Modal Analysis Model

The Automotive Exhaust Emission Modal Model[10] is a mathematical model developed to estimate light-duty vehicle emission data for carbon monoxide, hydrocarbons, and oxides of nitrogen over any specified driving sequence. The model is derived from data on emissions from 32 various driving conditions of the Surveillance Driving Sequences (SDS). Transient and steady state operating models were investigated in the SDS. The acceleration and deceleration modes were composed of 32 possible combinations of the following speeds: 0 mph, 15 mph, 30 mph, 45 mph, and 60 mph. Of these speeds, 32 modes were characterized by an average, constant acceleration and speed, while five steady state modes were established to complete the sequence.

The main accomplishment of the Modal Analysis Model was the development of a scheme which expanded the emissions from the 32 modes into a continuous function of time. The emission rates for all possible combinations of speed and acceleration may be calculated by this emissions model. This allows the analyst to estimate the emission rates for carbon monoxide, hydrocarbons,

and nitrogen oxides provided that the model is used within the speed and acceleration ranges spanned by the modal data.

C. Mass Balance Technique

The calculation of a pollutant emission factor for non-reactive species may be calculated by the use of concentration profiles at a downwind receptor. This technique was first investigated from air quality data collected by the Texas Transportation Institute under project 218.[11]

The process involves a numerical integration of the concentration fluxes passing a downwind receptor. The method assumes that both concentration and mass flux are only a function of height along any plane parallel to the roadway. The integrated area is then used along with traffic volumes to obtain a composite vehicular emission factor.

The mass balance technique suffers two disadvantages: (1) the emission factor may only be calculated for existing roads and (2) the analyst must have accurate air quality, traffic, and meteorological data to estimate the emission rate. However, the technique does allow a valid comparison to be made between mathematical models and actual air quality data.

D. MOBILE3

MOBILE3[12] is a third generation emissions model that calculates emission factors for hydrocarbons, carbon monoxide, and oxides of nitrogen for motor vehicles. The model calculates emission factors for eight different vehicle types in low and high altitude regions. The program estimates emission data for any calendar year between 1960 and 2020, inclusive. The emission factors calculated by MOBILE3 depend, in part, on ambient temperature, vehicle speed, mileage accrual rates, registration distribution, tampering with the emission control systems on automobiles, and such factors as trailer towing, air conditioning usage and inspection/maintenance programs.

Unlike its predecessors, MOBILE3 does not model emission rates for California scenarios. However, the program incorporates several new options, calculating methodologies, emission factor estimates, emission control regulations, and internal program designs. The calculational procedures used by MOBILE3 are presented in EPA publication AP-42.

E. TEXAS-II

TEXAS-II was developed at the Center for Transportation Research at the University of Texas at Austin[13] to predict pollutant emissions as well as the amount of fuel consumed by a vehicle passing through an intersection environment. Unlike most emission models, TEXAS-II was developed exclusively for intersection scenarios. The model calculates traffic behavior at intersections and applies the information to a quantitative evaluation of emissions and fuel consumption.

The model uses the following factors to determine emission data: (1) intersection size, (2) presence of a special left-turn lane, (3) pretimed signal control, (4) fully activated signal control, (5) all-way stop sign control, (6) traffic volumes, (7) left turns, and (8) heavy duty vehicles.

A modified version of the Texas Model for Intersection Traffic[14] is used to calculate momentary speed, acceleration, and position information. This information is required to calculate emissions and fuel consumption for light and heavy duty vehicles. The calculations are performed by use of an internal EPA Modal Analysis Model and a new model developed by Wu.[15]

II. Modeling Pollutant Dispersion Near Roadways

The development of a mathematical dispersion model to calculate the concentration of a pollutant in the vicinity of roadways has been approached in several ways. The most common dispersion models use one or more of the following approaches: (1) the gradient transport approach, (2) the statistical approach, (3) the similarity approach, and (4) the empirical approach. Pasquill[16] stated that the gradient transport approach is a mathematical development of a particular physical model of mixing. The second approach models the turbulent flow of material near the roadways in terms of the statistical properties of motion. In the similarity approach, postulations are formulated regarding the diffusion controlling physical parameters. Dimensional analysis is then used to relate those parameters to the dispersion process. The empirical approach uses a data base to develop empirical correlations relating concentration to a set of measured variables such as wind speed and direction.

Nearly all of the present roadway dispersion models use some form of the gradient transport approach, combined with empirical adjustments that are based on experimental data. Current models usually differ in the wide variety of assumptions used to solve the diffusion equations and the amount of empiricism incorporated into the development of the calculational procedure. Since this project is concerned with gradient transport models only, the reader is referred to Pasquill,[16] Sutton,[17] or Hanna *et al.*[18] for a review of other dispersion schemes.

A. General Atmospheric Diffusion

The equation which has been the basis for most gradient transport studies is the convective diffusion equation:

$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial x} + v\frac{\partial C}{\partial y} + w\frac{\partial C}{\partial z} = S + \frac{\partial}{\partial x}\left(K_x\frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y\frac{\partial C}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z\frac{\partial C}{\partial z}\right) \qquad (2-1)$$

where:

S = effects from all internal sources

C = concentration

t = time

x, y, z = the directions in the Euclidean coordinate system shown in Figure 1

u, v, w = wind velocity components in the u, v, and w directions, respectively

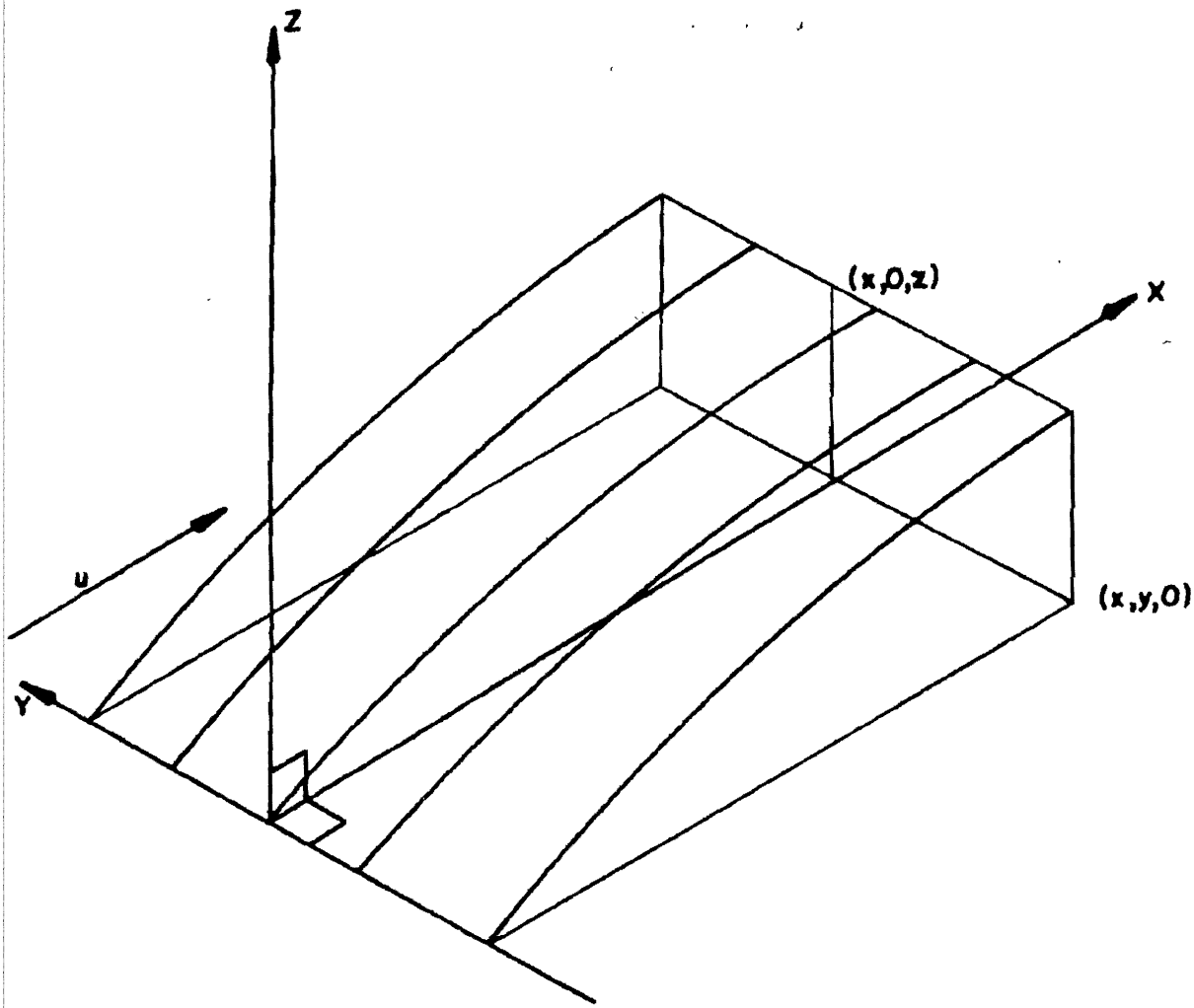


Figure 1

Euclidean Coordinate System

K_i = eddy diffusivity ($i = x, y, z$) in the x, y, and z directions, respectively

The cross diagonal terms, ($e.g.$, $\frac{\partial}{\partial x}K_{xy}\frac{\partial C}{\partial y}$) have been omitted in equation (2-1) because they are normally insignificant in atmospheric diffusion. Equation (2-1) may be termed as a continuity equation for pollutants in the atmosphere. Treybal[19] presents a derivation of such equations from the general continuity equation.

Most general solutions to equation (2-1) employ several general assumptions. If the assumptions of time independency, no net wind in the z–direction ($w = 0$), a perpendicular wind (in the x–direction), and $u\frac{\partial C}{\partial x} >> \frac{\partial}{\partial x}K_x\frac{\partial C}{\partial x}$, $i.e.$, advection dominates diffusion in the downward direction, are made, the equation can be reduced to:

$$u\frac{\partial C}{\partial x} = \frac{\partial}{\partial y}\left(K_y\frac{\partial C}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z\frac{\partial C}{\partial z}\right) \qquad (2-2)$$

The most general solutions to equation (2-2) found in the literature were presented by Smith.[20] These solutions assumed power law forms (in terms of z) for both wind speed and eddy diffusivity profiles. Smith derived solutions for both a ground level infinite line source and a point source. Solutions were also presented for the case of an elevated source.

A less general solution to equation (2-2) was derived by Roberts and published by Pasquill.[16] The assumed forms of the wind speed and eddy diffusivity profiles were more restricted than in Smith's solutions. Green[21] used this solution to develop the TRAPS-IIM model.

Sutton[17] presented solutions to both the ground level point source and infinite line source problems. These solutions assumed that the diffusion was Fickian. Fickian diffusion assumes constant eddy diffusivities. In the case of Fickian diffusion, equation (2-2) becomes

$$u\frac{\partial C}{\partial x} = K_y\frac{\partial^2 C}{\partial y^2} + K_z\frac{\partial^2 C}{\partial z^2} \qquad (2-3)$$

The solutions to the above partial differential equation proposed by Sutton are:

Point Sources:

$$C = \frac{Q}{4\pi x\sqrt{K_y K_z}}\exp\left[-\frac{\overline{u}}{4x}\left(\frac{y^2}{K_y} + \frac{z^2}{K_z}\right)\right] \qquad (2-4)$$

where:

Q = source strength

$\overline{u}$ = constant average wind speed

Infinite line sources:

$$C = \frac{Q'}{\sqrt{2\pi K_z x}}\exp\left(-\frac{\overline{u}z^2}{4K_z x}\right) \qquad (2-5)$$

where:

Q' = line source strength per unit length

The above solutions are based on a constant wind speed profile with height.

Since eddy diffusivities are not readily available, Gaussian solutions to the diffusion equation were proposed. Cramer[22] tested the following form of the Gaussian equation using data from Calder[23] and Barad:[24]

$$C = \frac{Q}{2\pi\sigma_y\sigma_z\bar{u}} \exp\left[-\frac{1}{2}\left(\frac{y^2}{\sigma_y^2} + \frac{z^2}{\sigma_z^2}\right)\right] \qquad (2-6)$$

where:

σ_y, σ_z = standard deviations of the concentration distribution in the y and z–directions, respectively.

Gifford[25] modified a set of dispersion curves originally presented by Pasquill[26] to predict the standard deviations as a function of downwind distance and atmospheric stability. These curves, commonly referred to as the Pasquill-Gifford curves, are routinely used in the estimation of σ_y and σ_z for Gaussian dispersion models.

Sutton[17] presented an argument for modifications to the Gaussian dispersion equations based on the ground being impervious to pollutant. This argument led to the variable source height solution:

Point source:

$$C = \frac{Q}{2\pi\sigma_y\sigma_z\bar{u}} \exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_y}\right)^2\right]\left\{\exp\left[-\frac{1}{2}\left(\frac{z-h}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+h}{\sigma_z}\right)^2\right]\right\} \qquad (2-7)$$

Infinite line source:

$$C = \frac{Q'}{\sqrt{2\pi}\sigma_z\bar{u}}\left\{\exp\left[-\frac{1}{2}\left(\frac{z-h}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+h}{\sigma_z}\right)^2\right]\right\} \qquad (2-8)$$

where:

h = source height

The last terms in equations (2-7) and (2-8) account for reflection of the plume at the ground by assuming an image source at distance h below the ground.

Most existing atmospheric models use some form of equations (2-7) or (2-8) to model the pollutant dispersion. In this report, the dispersion models HIWAY-2, CALINE4, and TXLINE will be discussed.

B. HIWAY-2

HIWAY-2 was developed by Petersen[4] and is a revised version of the EPA model HIWAY.[27] The model takes each lane of traffic as a line source of finite length. By summing concentration predictions from separate finite line segments, the model may be used for intersection scenarios. The

model uses Gaussian dispersion equations similar to those presented by Turner.[28] Concentration estimates are made by numerical integration of the appropriate point source equation:

$$C = \frac{Q'}{u} \int_0^D f \, dl \qquad (2-9)$$

where:

Q' = line source strength per unit length (g/m · sec)
u = wind speed (m/sec)
f = point source dispersion function (m^{-2})
C = pollutant concentration (g/m^3)
D = line source length (m)
l = length along line sources (m)

The model may be used with a wind speed estimate at about 2 meters height above the ground in relatively open terrain.

The point source dispersion function, f, takes different forms depending on mixing height and atmospheric stability. For stable conditions, or if the mixing height is greater than 5000 meters:

$$f = \frac{1}{2\pi\sigma_y\sigma_z} \exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_y}\right)^2\right] \left\{\exp\left[-\frac{1}{2}\left(\frac{z-h}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+h}{\sigma_z}\right)^2\right]\right\} \qquad (2-10)$$

where:

z = receptor height above ground (m)
h = effective source height (m)

In an unstable or neutral atmosphere, if σ_z is at least 1.6 times the mixing length L:

$$f = \frac{1}{\sqrt{2\pi}\sigma_y L} \exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_y}\right)^2\right] \qquad (2-11)$$

In all other unstable or neutral conditions:

$$\begin{aligned} f = \frac{1}{2\pi\sigma_y\sigma_z} \exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_y}\right)^2\right] &\left\{\exp\left[-\frac{1}{2}\left(\frac{z-h}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+h}{\sigma_z}\right)^2\right]\right. \\ &+ \sum_{n=1}^{\infty}\left[\exp\left[-\frac{1}{2}\left(\frac{z-h-2nL}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+h-2nL}{\sigma_z}\right)^2\right]\right. \\ &\left.\left. + \exp\left[-\frac{1}{2}\left(\frac{z-h+2nL}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+h+2nL}{\sigma_z}\right)^2\right]\right]\right\} \end{aligned} \qquad (2-12)$$

Petersen claims that the infinite series in equation (2-12) converges rapidly and usually no more than four or five sums of the four terms are required. In all cases, σ_y and σ_z are evaluated for the given stability class and downwind distance.

The integral in equation (2-9) is evaluated using the Richardson extrapolation of the trapezoidal rule. Concentration estimates are first made by dividing the line segment into a number of intervals equal to $3, 6, \ldots, 3 \times 2^9$. (Each interval is represented by a point source.) Calculations are repeated successively until the concentration estimates converge to within 2% of the previous estimate. If convergence is not obtained by the time the number of intervals is 3×2^9, the estimated integral is saved and a sequence of new estimations for intervals equal to $4, 8, \ldots, 4 \times 2^9$ is performed. Any new integral estimate for interval values of $4, 8, \ldots, 2048$ (4×2^9) having a relative error of less than 2% from the saved integral signals convergence. The above integration is repeated for each lane of traffic and summed to represent the total concentration from the highway segment.

HIWAY-2 is capable of estimating pollutant concentration at locations downwind of a depressed highway (cut section). This is done by considering the top of the cut section to be an area source of pollution. This area source is approximated using ten line sources located at the top of the depressed section. The total emission rate of the roadway is first found by adding together the emission rates for each individual lane of traffic. This emission rate is then equally distributed over each of the ten line sources simulating the area source at the top of the cut section.

HIWAY-2 includes several empirical correlations to calculate the dispersion parameters. Several authors (Rao, *et al.*[29] and Eskridge, *et al.*[30,31]) have suggested that induced turbulence near roadways due to the traffic may play a larger role in pollutant dispersion than atmospheric stability. Therefore, the dispersion parameters are composed of a combination of atmospheric turbulence plus initial induced dispersion. HIWAY-2 incorporates an aerodynamic drag factor that accounts for the initial dispersion of the pollutant near the roadway. This allows the model to make a reasonable prediction of pollutant concentrations when the wind speed is low.

C. CALINE4

CALINE4 is a fourth generation dispersion model developed by Benson[3] for the California Department of Transportation. The model is based on the Gaussian solution to the diffusion equation and employs a mixing zone concept to characterize pollution dispersion near the roadway. The model is capable of predicting inert material concentrations, *i.e.*, carbon monoxide, nitrogen dioxide, and suspended particle concentrations within 500 meters downwind of a roadway. The model also contains intersection, street canyon, and parking facility capabilities.

CALINE4 divides the individual links in the roadway into a series of elements (Figure 2). Incremental concentration contributions are then calculated for each element and then summed to obtain the total downwind concentration at a given receptor. The downwind receptor distance is measured along a line perpendicular to the link centerline. The elements are not all of the same size; elements that are further away from the receptor are larger than those at small downwind distances. Each element is modeled as an equivalent finite line source (FLS) positioned normal

to the wind direction and centered at the element midpoint. All emissions resulting from each element are assumed to be released along the FLS. Emissions are distributed within each element by dividing each element into three sub-elements. The emission rate in the central sub-elements is assumed to be uniform across the sub-element. Emissions for the other sub-elements are assumed to decay linearly to zero at the element boundary.

Downwind contributions from each element are calculated by the crosswind FLS Gaussian equation:

$$dC = \frac{Q'\,dy}{2\pi u \sigma_y \sigma_z}\left[\exp\left(-\frac{y^2}{2\sigma_y^2}\right)\right]\left\{\exp\left[-\frac{(z-h)^2}{2\sigma_z^2}\right] + \exp\left[-\frac{(z+h)^2}{2\sigma_z^2}\right]\right\} \qquad (2-13)$$

where:

$dC =$ incremental concentration

Benson states the solution to equation (2-13) in the following form:

$$C = \frac{Q'}{\sqrt{2\pi}\sigma_z u}\left\{\exp\left[-\frac{(z-h)^2}{2\sigma_z^2}\right] + \exp\left[-\frac{(z+h)^2}{2\sigma_z^2}\right]\right\} \cdot PD \qquad (2-14)$$

where:

$$PD = \frac{1}{2\pi}\int_{y_1/\sigma_y}^{y_2/\sigma_y} \exp\left(-\frac{p^2}{2}\right) dp \qquad (2-15)$$

$p = y/\sigma_y$

$y_1, y_2 =$ endpoints of the FLS

Note that PD is the normal probability density function integrated over the length of the FLS. The model determines receptor concentration as a series of incremental contributions from each element FLS. Horizontal dispersion effects are determined by equation (2-15) which by definition is the area under the normal curve with standard deviation σ_y. Each FLS is divided into segments of length equal to a multiple of σ_y. The source strength for each segment is determined by use of the elemental lineal source strength and a weighting factor that considers the linear decrease of emissions across the peripheral sub-elements. The total receptor concentration is therefore:

$$C = \frac{1}{\sqrt{2\pi}u}\sum_{i=1}^{n}\left\{\frac{1}{SGZ_i}\sum_{k=-CNT}^{CNT}\left[\exp\left[-\frac{(z-h+2kL)^2}{2SGZ_i^2}\right] + \exp\left[-\frac{(z+h+2kL)^2}{2SGZ_i^2}\right]\right] \times \sum_{j=1}^{6}(WT_j \times QE_i \times PD_{ij})\right\} \qquad (2-16)$$

where:

$n =$ total number of elements

$CNT =$ number of reflections required for convergence

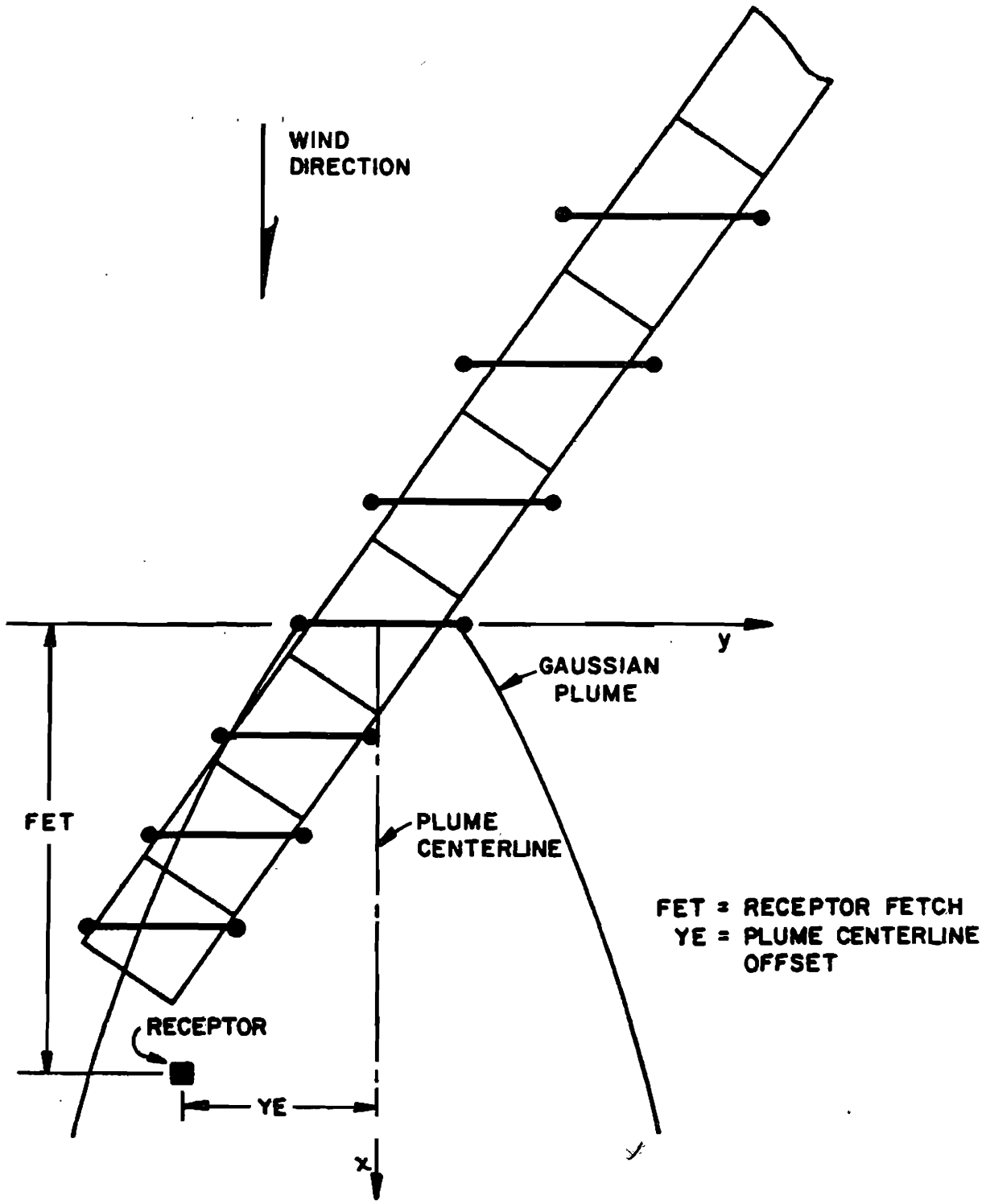


Figure 2

CALINE4 Roadway Treatment[3]

u = wind speed

L = mixing height

SGZ_i = σ_z as function of downwind distance for the ith element

QE_i = central sub-element lineal source strength for the ith element

WT_j = source strength weighting factor for the jth FLS segment (Accounts for linear decrease of emissions across the outer sub-elements)

$PD_{ij} = \frac{1}{\sqrt{2\pi}} \int_{\frac{Y_j}{SGY_i}}^{\frac{Y_{j+1}}{SGY_i}} \exp\left(-\frac{p^2}{2}\right) dp$

Y_j, Y_{j+1} = offset distances for the jth FLS segment

SGY_i = σ_y as a function of downwind distance for the ith element

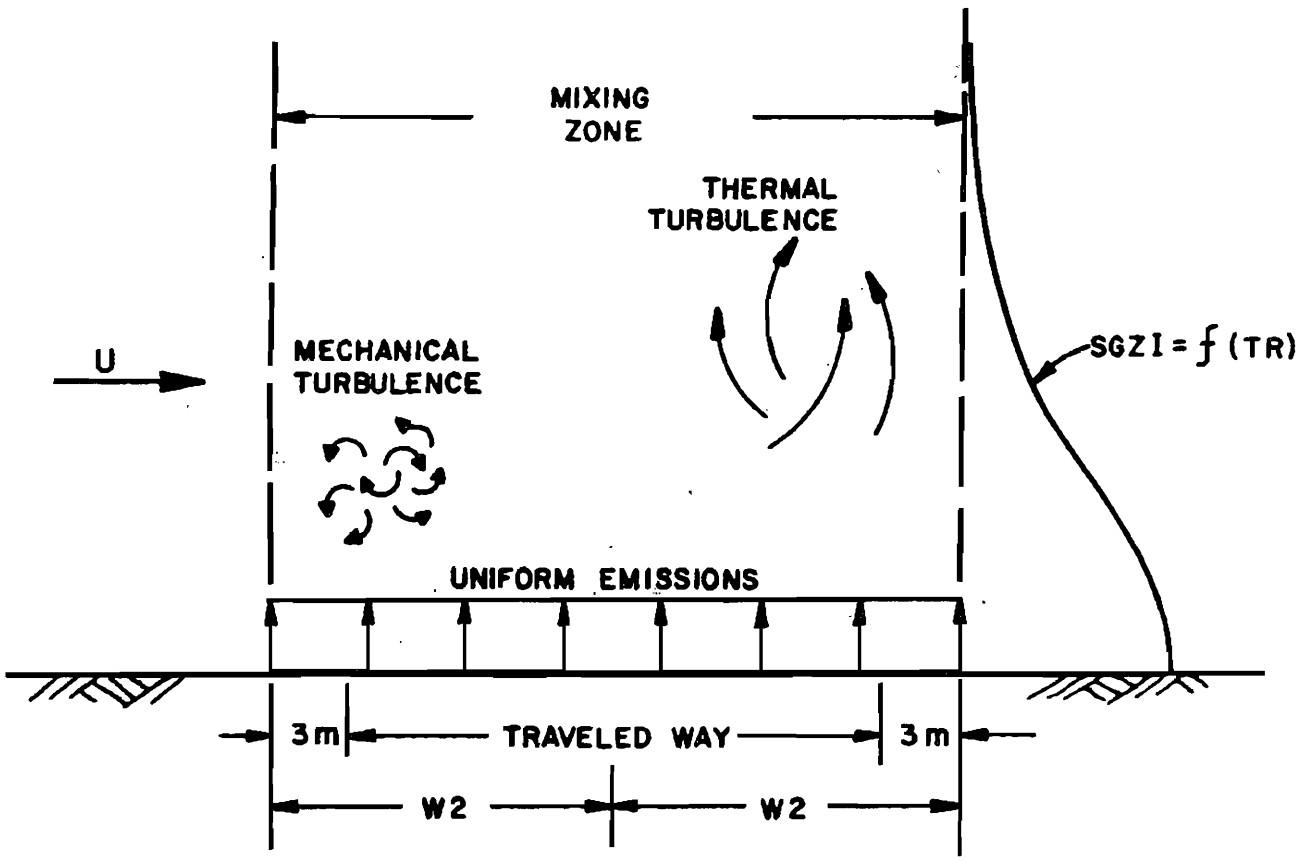
CALINE4 treats the zone directly above the roadway as a zone of uniform emissions and turbulence (Figure 3). This designated mixing zone is equal to the total width of the traveled lanes plus three meters on each side. These three meters are used to characterize the initial horizontal dispersion imparted to the pollutant by the moving traffic. Within this zone, the dispersion mechanisms are assumed to be dominated by thermal turbulence from the hot vehicle exhaust and mechanical turbulence from the vehicular wakes. Since this mechanical turbulence is not normally accounted for in standard Gaussian models, CALINE4 models the initial vertical dispersion parameter as a function of pollutant residence time in the mixing zone. A modified form of the Pasquill-Smith vertical dispersion curves are used to characterize the vertical dispersion parameter σ_y downwind from the mixing zone. This version of the curves takes in consideration the thermal effects of hot vehicular exhausts. The horizontal dispersion parameter σ_y is evaluated by a method proposed by Draxler.[32]

CALINE4 permits the user to specify a maximum of 20 links and 20 receptors. The model can be used to estimate dispersion from multiple sources, curved alignments, or roadway segments with differing emission factors. However since the model is based on the assumptions of horizontal homogeneous wind flow and steady-state meteorology, CALINE4 should be used with care on complex topography.

D. TXLINE

The original TXLINE model was developed at the Texas Transportation Institute in 1980.[2] The TXLINE model assumes a non-Fickian solution to the diffusion equation (2-2). The model is the successor to TRAPS-IIM.[21] The program estimates the emissions as an infinite line source. This limited the use of TXLINE from being used to model complex geometries or intersections. TXLINE is capable of modeling moderate cut and fill sections (2:1 grade or less) as well as elevated portions (bridges) of the roadway.

TXLINE utilizes the most general solutions to equation (2-2) derived by Smith.[20] For an



SGZI = INITIAL VERTICAL DISPERSION PARAMETER
TR = MIXING ZONE RESIDENCE TIME

Figure 3

Turbulent Emission Zone of CALINE4[3]

infinite line source,

$$C = \frac{Q'\left[h(z-h)+h^2\right]^{m/2}}{(1+2m)K_1 z}\exp\left[\frac{u_1 z^{1+2m}+u_1 h^{1+2m}}{K_1(2m+1)^2 x}\right] \times I_{-m/(1+2m)}\frac{2u_1\left[(z-h)h+h^2\right]^{(1+2m)/2}}{K_1(2m+1)^2 x} \qquad (2-17)$$

where:

$I_{-\frac{m}{1+2m}}$ = Bessel function of the first kind with order equal to $-m/(1+2m)$

m = power law wind speed parameter from $u(z) = u_1\left(\frac{z}{z_1}\right)^m$

u_1 = reference wind speed at height $z = z_1$

K_1 = eddy diffusivity at height $z = z_1$

z_1 = reference height taken as 1 meter

x = downwind distance (in wind direction)

As a special case of equation (2-17), the infinite ground level line source equation may be formulated ($h = 0$):

$$C = \frac{Q'\left(x\frac{K_1}{u_1}\right)^{-\frac{1+m}{1+2m}}}{u_1(1+2m)^{\frac{1}{1+2m}}\left(-\frac{m}{1+2m}\right)!}\times\exp\left[-\frac{u_1 z^{1+2m}}{K_1(1+2m)^2 x}\right] \qquad (2-18)$$

where:

$-\frac{m}{1+2m}!$ = is a factorial of a non-integer. This value is calculated by use of the gamma function: $\Gamma(1-\frac{m}{1+2m}) = -\frac{m}{1+2m}!$

Due to the complexity of the equations, the solution of the elevated point source problem has not been found yet for general values of m. However, Smith was able to derive a solution for the special case of $m = \frac{1}{2}$. This solution gave him a valuable hint for determining the form of the ground level point source solution. For the ground source point solution, Smith gives:

$$C = X(x,z)\exp\left(-\frac{y^2}{f(x,z)}\right) \qquad (2-19)$$

where:

$f = 2C_2/C_o$

$X = C_o\sqrt{(C_o/2\pi C_2)}$

$C_o = \int_{-\infty}^{\infty} C\,dy$

$C_2 = \int_{-\infty}^{\infty} y^2 C\,dy$

C_o is the solution to the infinite ground level line source equation and C_2 is a measure of the spread in the $y-$direction. TXLINE uses a simplified version of the spread function as given by Smith.[20]

It also assumes that the eddy diffusivities in the y and $x-$directions are equal. The spread function C_2 is given by:

$$C_2 = \frac{2Q'K_1^{b-a}}{u_1^{b-a+1}}(1+2m)^{(3b-4)/2}\left[\frac{(b-1)!\,(b+a-2)!}{(a-1)!\,(2b-1)!}\right]x^{b-a} \times e^{-\eta}\left[\frac{(b-1)!}{(a-1)!}{}_1F_1(b;\,a;\,\eta) - \eta^b V(b;\,a;\,\eta)\right] \qquad (2-20)$$

where:

$$\eta = \frac{u_1 x^{1+2m}}{(1+2m)^2 K_1 x}$$
$$a = (1+m)/(1+2m)$$
$$b = 2/(1+2m)$$

The solution contains two rapidly converging series: ${}_1F_1$, known as Kummer's function,

$${}_1F_1(b;\,a;\,\eta) = 1 + \frac{b\eta}{a} + \frac{(b)_2\eta^2}{(a)_2 2!} + \ldots + \frac{(b)_n\eta^n}{(a)_n n!} \qquad (2-21)$$

where:

$$(a)_n = a(a+1)(a+2)(a+3)\ldots(a+n-1)$$
$$(a)_0 = 1$$
$$(b)_n = b(b+1)(b+2)(b+3)\ldots(b+n-1)$$
$$(b)_0 = 1$$

and an allied function, V:

$$V(b;\,a;\,\eta) = \sum_{r=0}^{\infty}\frac{(2b+r-1)!}{(b+r)!\,(b+a+r-1)!}\eta^r \qquad (2-22)$$

All factorials in TXLINE are calculated by use of the gamma function.

Most roadway models assume that the dispersion process is Gaussian in both the crosswind and vertical directions. The equations derived by Smith used in TXLINE illustrate that the model assumes a Gaussian distribution in the $y-$direction. However, the dispersion in the $z-$direction is definitely not Gaussian.

For wind directions within twenty degrees of the roadway, TXLINE uses the infinite line source solution. For all other cases, the program reduces the infinite line source problem to a numerical integration of the point source equation. The roadway is represented by a series of closely spaced point sources. Each point source is oriented with its $x-$axis parallel to the wind. Before integration is performed, a coordinate transformation is required. Each point source coordinate system has a different origin therefore, the x and $y-$coordinates of a given receptor depend on the location of the point source. The transformations used are:

$$x' = x\sin\theta - p\cos\theta \qquad (2-23a)$$

$$y' = x\cos\theta - p\sin\theta \qquad (2\ \ 23b)$$

where:

$\theta =$ wind angle with line source (0° = parallel, 90° = perpendicular)

$x' =$ $x-$coordinate of the receptor with respect to the point source coordinate system

$y' =$ $y-$coordinate of the receptor with respect to the point source coordinate system

$x =$ $x-$coordinate of the receptor with respect to the line source (the line source is the $p-$axis)

$p =$ $p-$coordinate of the point source

These points are shown in Figure 4. The concentration profile downwind of a ground level infinite line source for any general wind angle is defined by replacing the variables x and y in the ground level point source equation by the transformed variables x' and y', and integrating from $-\infty$ to $+\infty$ with respect to the $p-$direction (along the line source). The profile is then:

$$C = \int_{-\infty}^{\infty} C_o\sqrt{\frac{C_o}{2C_2}}\exp\left[-\frac{C_o(x\cos\theta + p\sin\theta)^2}{2C_2}\right] dp \qquad (2-24)$$

where:

C_o is defined by equation (2-18) with x replaced by x'

C_2 is defined by equation (2-20) with x replaced by x'

Power law forms for the eddy diffusivities and the wind profile are used by the model and are given by:

$$K(z) = K_1\left(\frac{z}{z_1}\right)^{1-m} \qquad (2-25a)$$

$$u(z) = u_1\left(\frac{z}{z_1}\right)^{m} = u_* q\left(\frac{z}{z_0}\right)^{m} \qquad (2-25b)$$

where:

$K_1 =$ eddy diffusivity at 1 meter

$u_* =$ friction velocity

$u_1 =$ reference wind speed at 1 meter

$z_0 =$ surface roughness

Polynomials that express q and m as a function of z_0 are given in TTI report TTI-2-8-80-283-1.[2] For $0 < z_0 \leq 0.30$ m:

$$m = 0.143 + 1.901z_0 - 15.62{z_0}^2 + 83.24{z_0}^3 - 224.4{z_0}^4 + 236.0{z_0}^5 \qquad (2-26a)$$

$$q = 5.818 - 46.12z_0 + 416.4{z_0}^2 - 2162.3{z_0}^3 + 5671.0{z_0}^4 - 5830.0{z_0}^5 \qquad (2-26b)$$

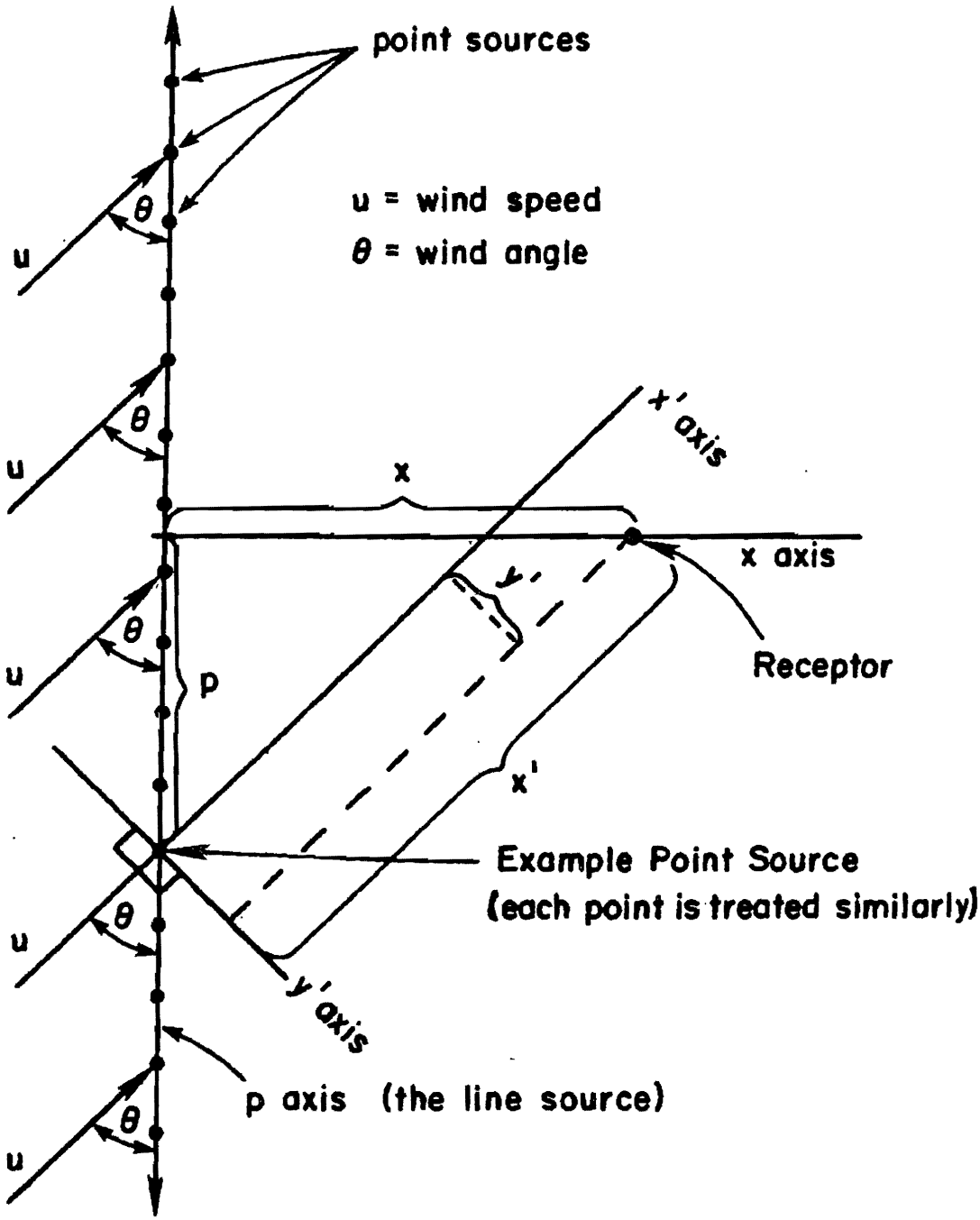


Figure 4

TXLINE Point Source Representation of a Line Source

For General Wind Angles

For $0.30 < z_0 < 0.50$ m:

$$m = 0.229 + 0.306z_0 - 0.122{z_0}^2 + 0.040{z_0}^3 - 0.0066{z_0}^4 + 0.0004{z_0}^5 \quad (2-27a)$$

$$q = 3.827 - 4.385z_0 + 4.50{z_0}^2 - 2.88{z_0}^3 + 1.102{z_0}^4 - 0.245{z_0}^5 + 0.029{z_0}^6 - 0.0014{z_0}^7 \quad (2-27b)$$

With these expressions, u_* can be calculated from the log wind profile equation:

$$u = \frac{u_*}{k} \ln\left(\frac{z}{z_0}\right) \quad (2-28)$$

where:

$k =$ 0.4 (known as von Kármán's constant)

From a knowledge of u_*, equation (2-25b) can then be used to determine u_1. The eddy diffusivity K_1 can then be calculated from the relationship:

$$K_1 = \frac{u_1 z_0^{2m}}{mq^2} \quad (2-29)$$

The TXLINE model uses a wind speed correction factor to improve performance at low wind speeds. The correction factor is a function of wind angle with respect to the road and wind speed. The factor will thus increase the supplied wind speed to improve model results.

E. Composite Models

For intersection analysis, several models have been developed which utilize combinations of the preceding models and assorted traffic engineering principles to predict pollutant concentrations. A three step approach is commonly applied to many of these models which involves the processes of:

(1) Traffic flow analysis
(2) Calculation of intersection emissions
(3) Modeling of the dispersed pollutants

The Intersection Midblock Model (IMM), MICRO, and TEXIN are three composite models which are briefly described below.

Intersection Midblock Model (IMM)

In accordance with the above mentioned three step process, the IMM[6] combines accepted traffic engineering principles with the Modal Analysis Model, MOBILE1, and HIWAY-2 programs to calculate carbon monoxide concentrations near intersections. It is designed as a screening tool to identify potential urban *hot spots*, or areas of high pollutant concentrations.

MICRO

The Colorado Department of Highways developed the program MICRO to determine the impact of traffic signalization on air quality.[7,8] The three step procedure is accomplished through

the use of the intersection submodel of the regional air quality dispersion model, APRAC-2,[33] a modified Modal Analysis Model, and dispersion equations similar to those found in HIWAY-2.

The emissions are calculated along each link of the intersection and are then modeled by a Gaussian point source formulation. The links are subdivided into several smaller sections, each being modeled as a separate point source. The contributions from the links are summed to give the pollutant concentration at the selected receptor.

TEXIN

The TEXIN model was developed at Texas A&M University by Nelli, *et al.*[34] The program follows the common three-step procedure by implementing various traffic algorithms, the MOBILE2 emissions model, and the CALINE3 dispersion model. The general flow diagram is presented in Figure 5.

In addition to simple signalized and unsignalized intersections with four straight legs, the TEXIN model is capable of handling more complex situations, including curved roadways. Minor intersections can also be modeled concurrently with the major intersection, provided that the side street is controlled by a stop or yield sign. The model is not applicable, however, to *street canyon* configurations.

III. Collection of Experimental Data Bases

The development of accurate vehicle dispersion models cannot be effected without the availability of reliable data bases. These data must include a wide variety of measurements over varying meteorological conditions to be useful. Data on meteorological conditions, traffic volumes, and pollutant concentrations must be collected. Over the past 15 years, several comprehensive data bases have been released. This section focuses on four data bases collected by independent groups. A list of references for other data bases is included at the end of this section.

A. General Motors Dispersion Experiment

The General Motors Dispersion experiment was performed at the GM proving grounds in Milford, Michigan, and is discussed in detail by Cadle, *et al.*[35] The EPA and other governmental agencies assisted in the planning and execution of the experiment. The study measured the dispersion of a sulfur hexafluoride (SF_6) tracer gas, as well as the diffusion of particles and sulfate. The SF_6 tracer gas experiments were of primary interest since they led to several correlations used in many popular dispersion models (CALINE4, HIWAY-2, and TXLINE).

The test site was a 10 km straight track (5 km in each direction) with banked turns at each end. A fleet of 352 cars were used to simulate roadway traffic. The drivers would maneuver the cars in 32 packs of 11 cars at a constant speed of 80 km/hr. Seven or eight light duty trucks were equipped with cylinders of SF_6 tracer gas that could be released into the atmosphere at a known rate. The release vehicles were evenly distributed among the entire fleet.

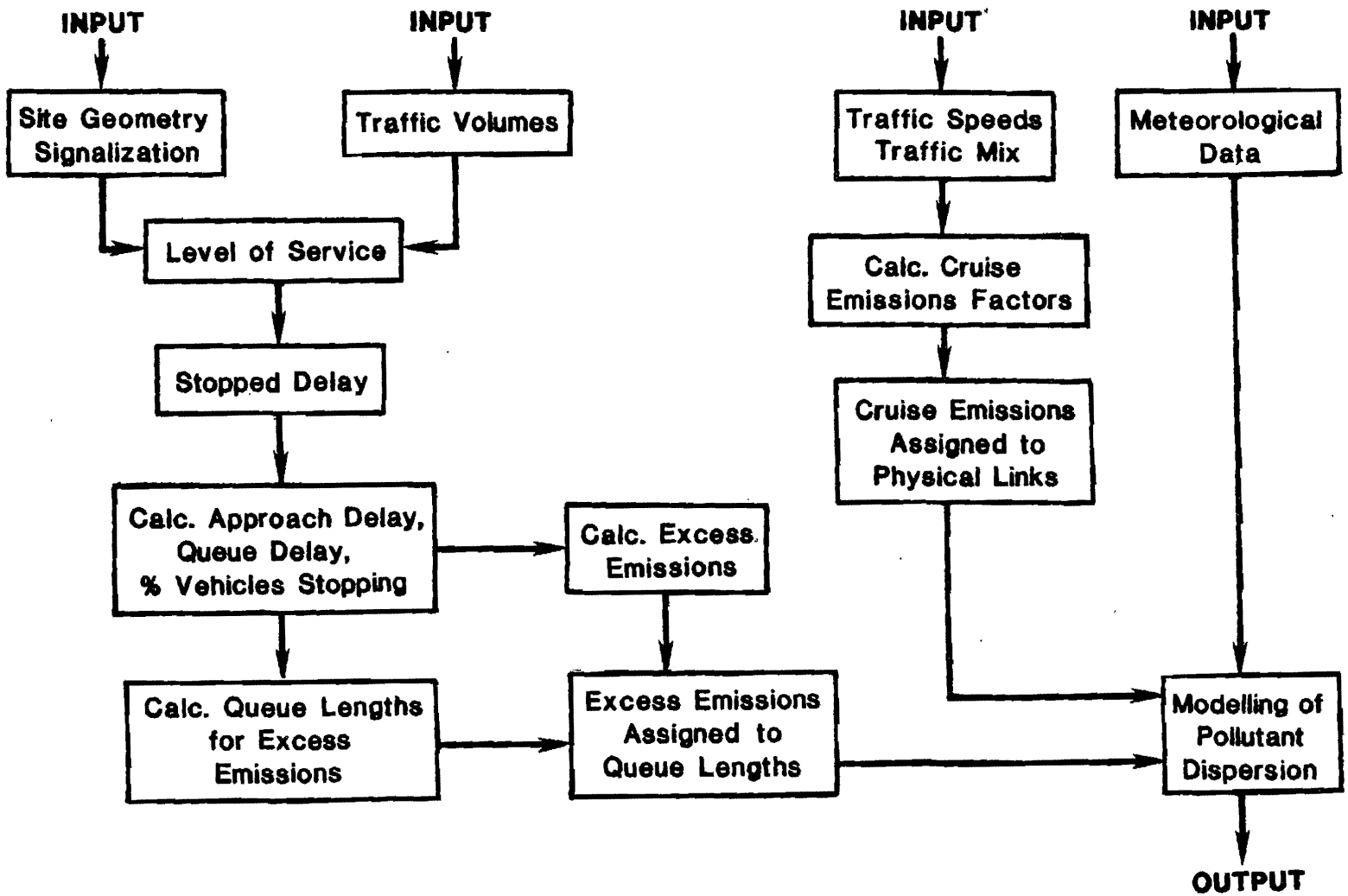


Figure 5

Flow Diagram for the Texas Intersection (TEXIN) Model

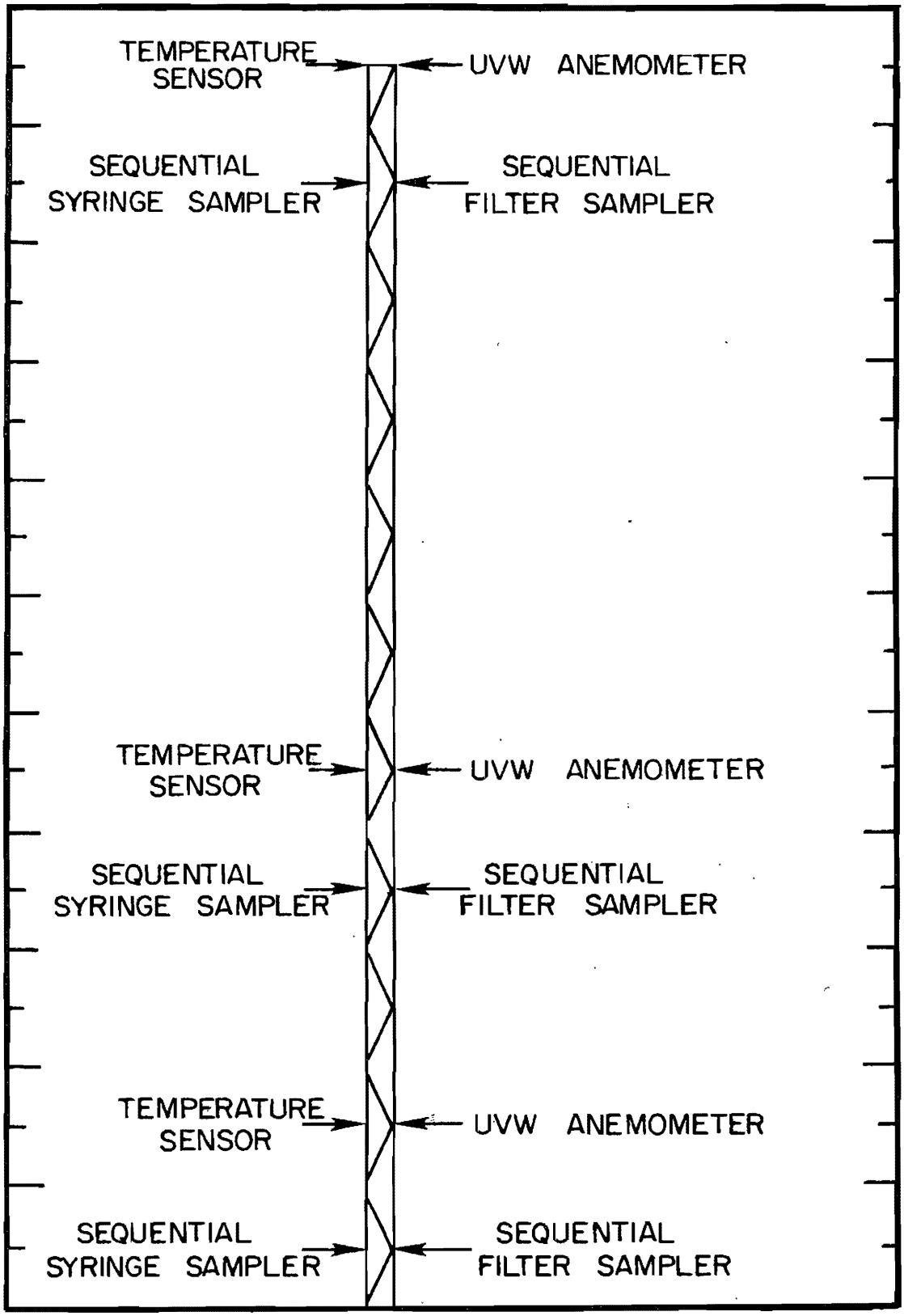


Figure 7

Tower Instrumentation in GM Experiment[35]

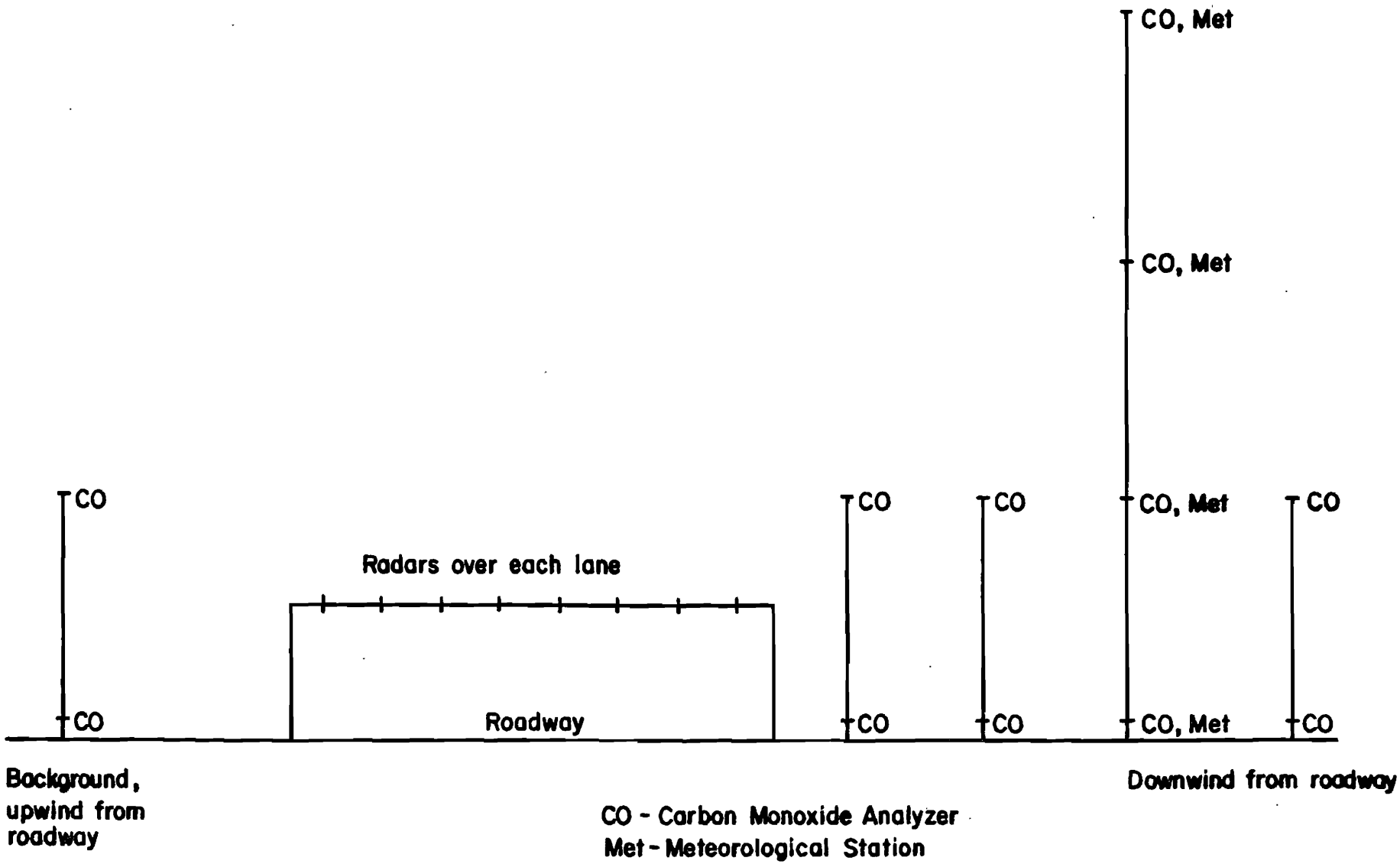


Figure 8

General Instrumentation Layout for the Texas A&M Data Base

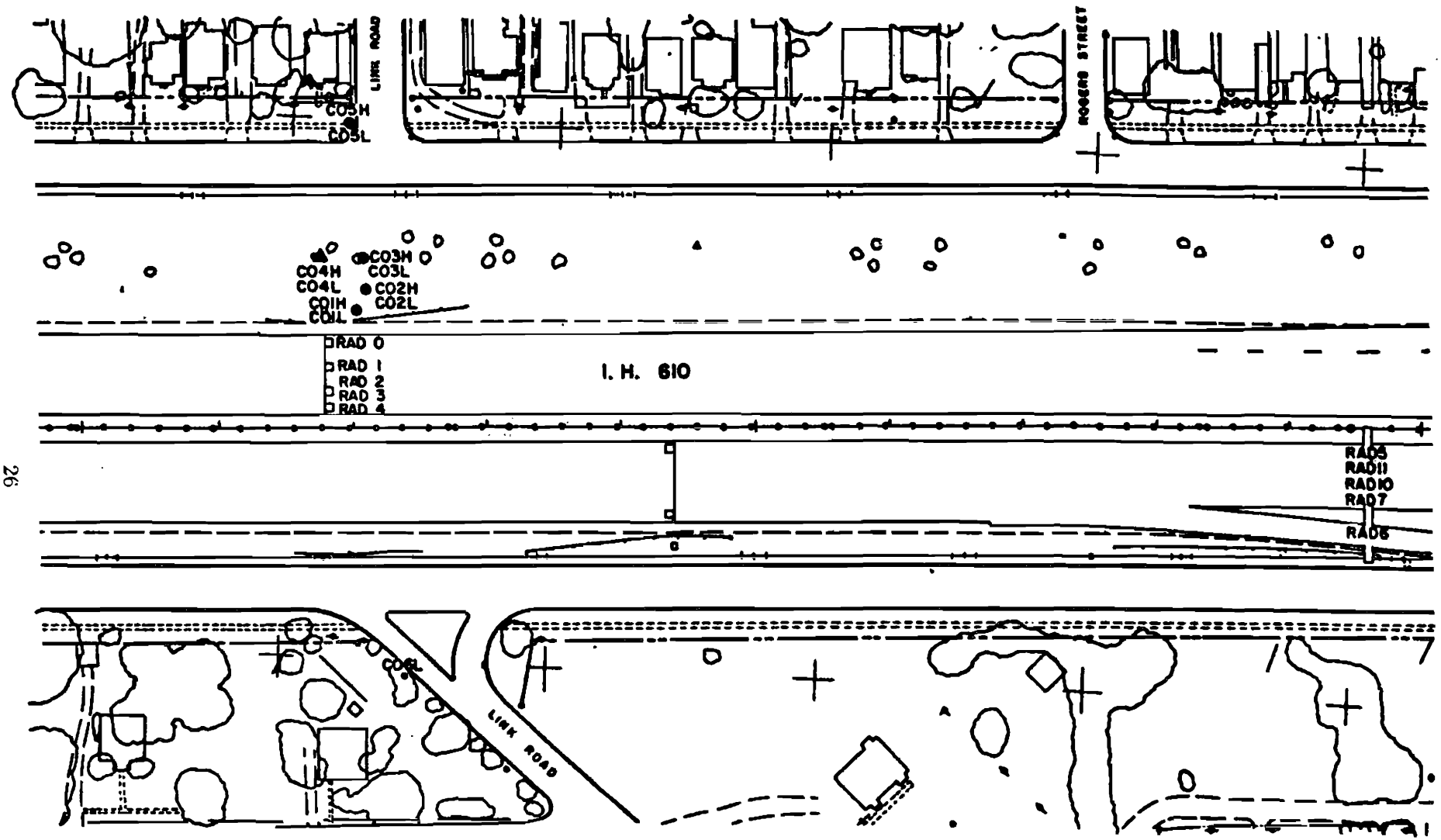


Figure 9

Overhead View of the Houston at-Grade Site—IH610 at Link Road

INSTRUMENT LOCATIONS

① = VA 1.5M, HA 1.5M, TM 1.5M, WV 1.5M, RH 1.5M
② = VA 10M, HA 10M, TMP 10M, WV 10M
③ = VA 20M, HA 20M, TMP 20M, WV 20 M
④ = TMP 30 M, RH 30 M
⑤ = VA 40M, HA 40 M, WV 40 M

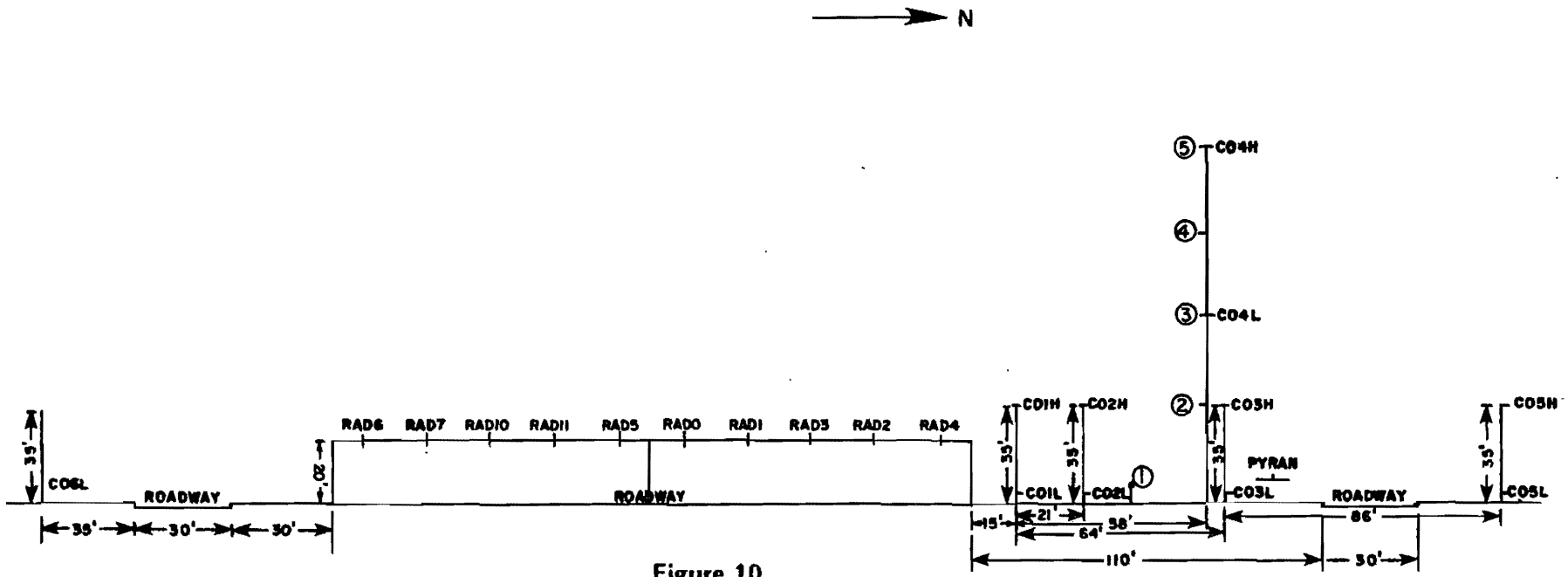


Figure 10

Cross Section of the Houston at-Grade Site—IH610 at Link Road

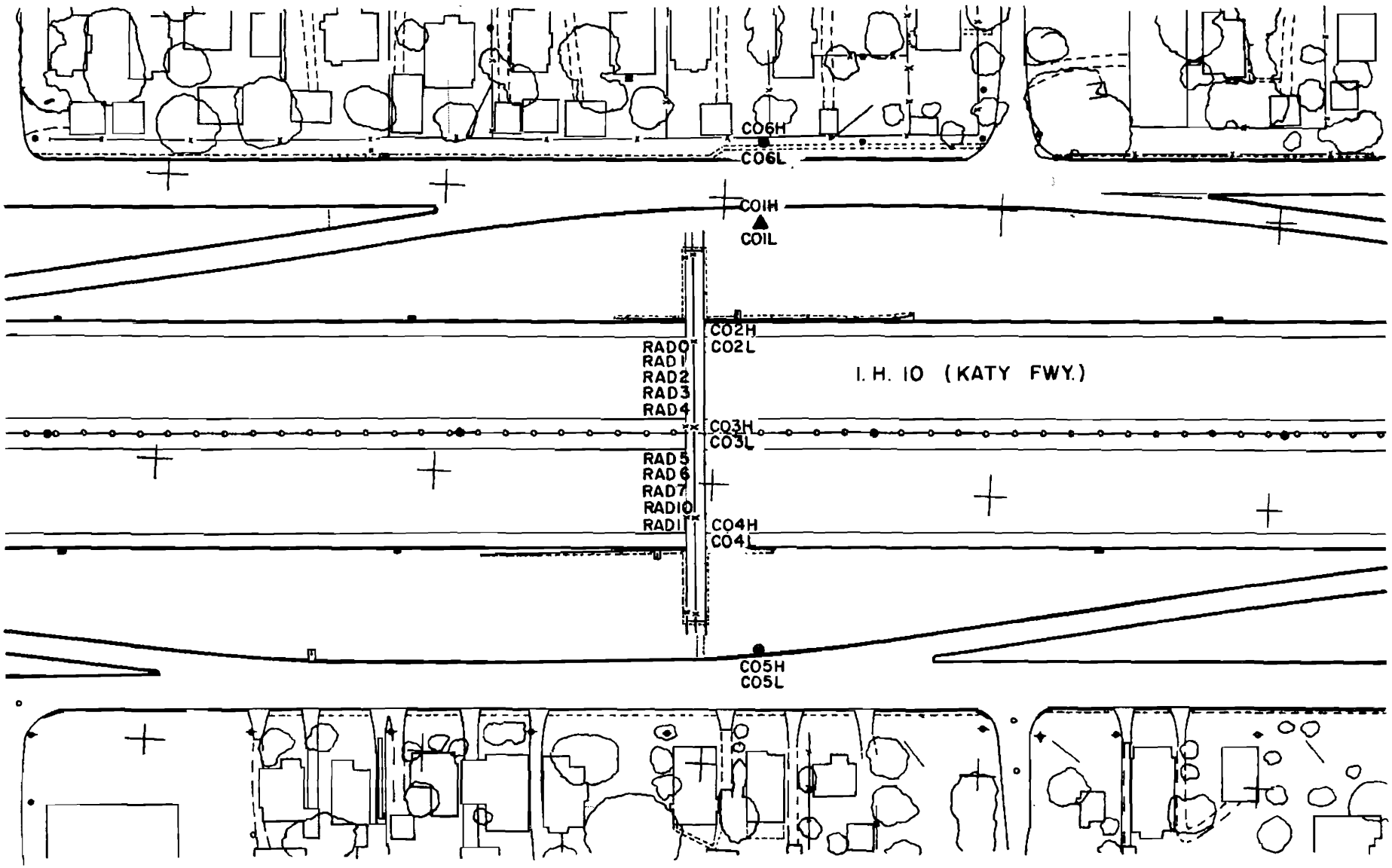


Figure 11

Overhead View of the Houston Cut Site—IH10 at Reinerman Road

INSTRUMENT LOCATIONS

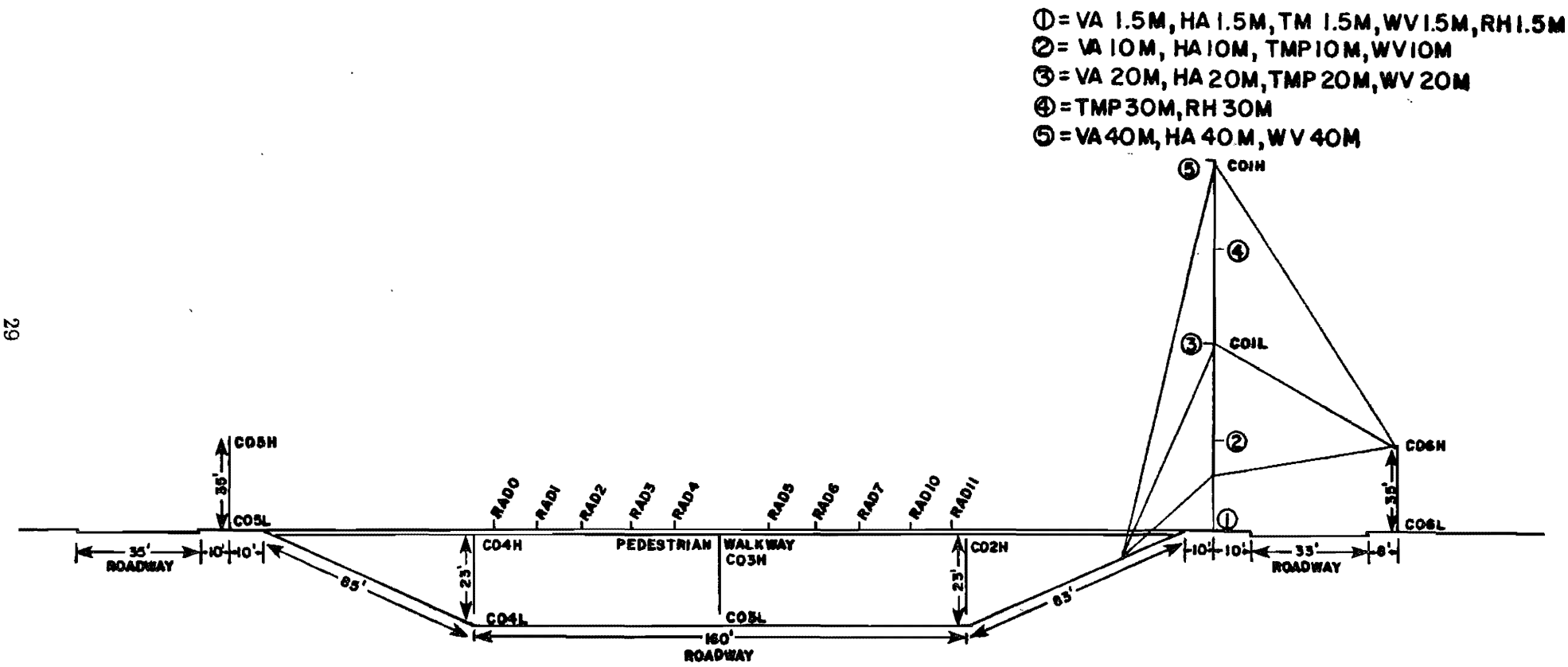


Figure 12

Cross Section of the Houston Cut Site—IH10 at Reinerman Road

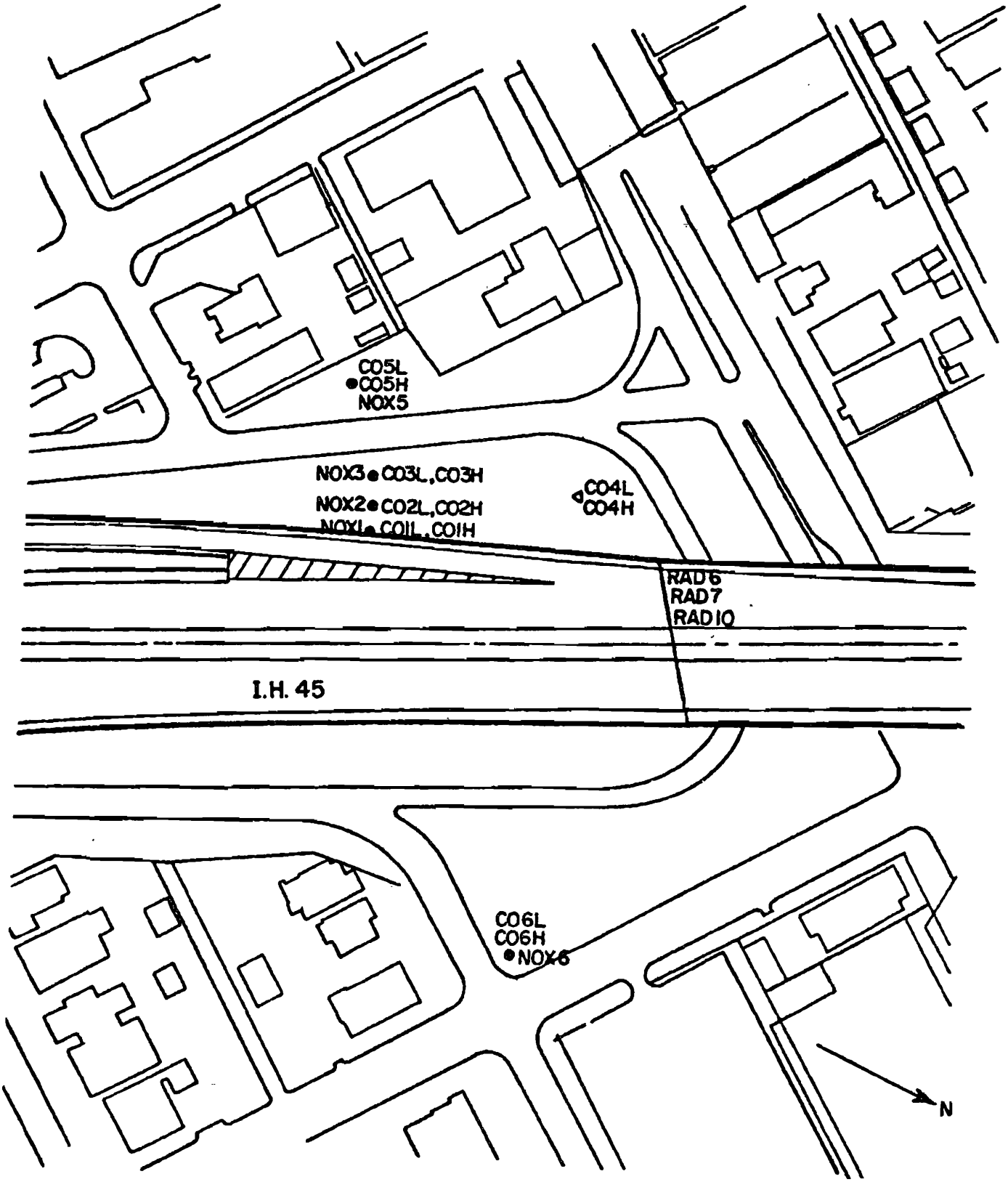


Figure 13

Overhead View of Dallas Elevated Site

IH45 at Forest Avenue

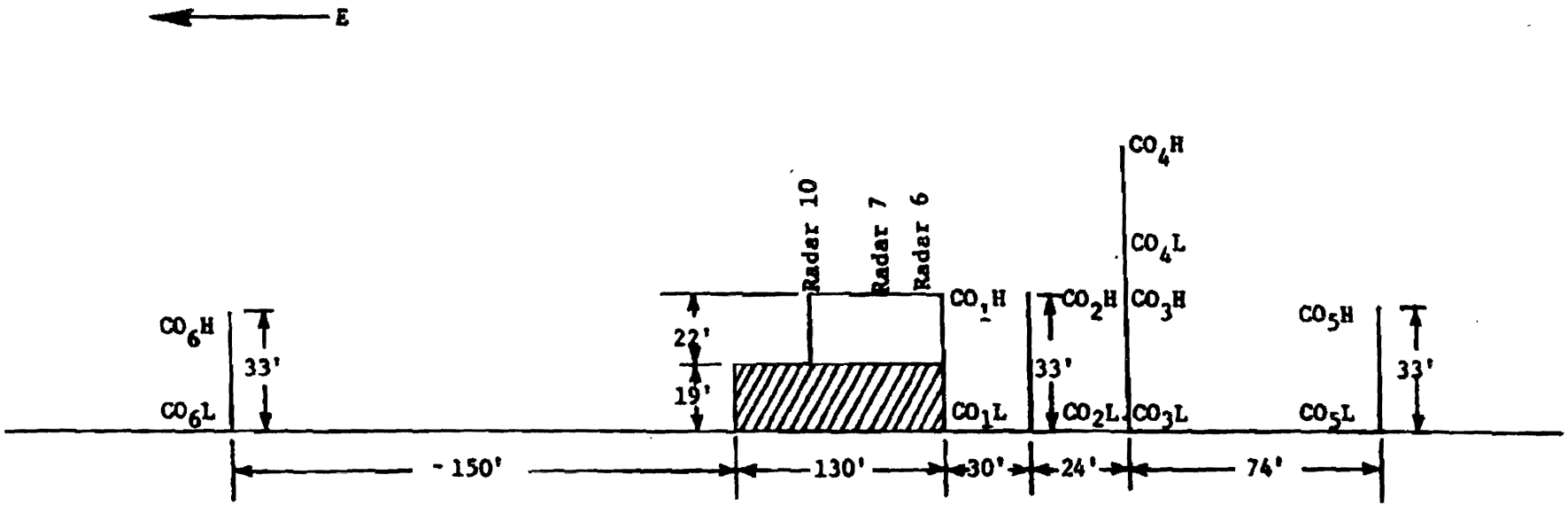


Figure 14

Cross Section of the Dallas Elevated Site—IH45 at Forest Ave.

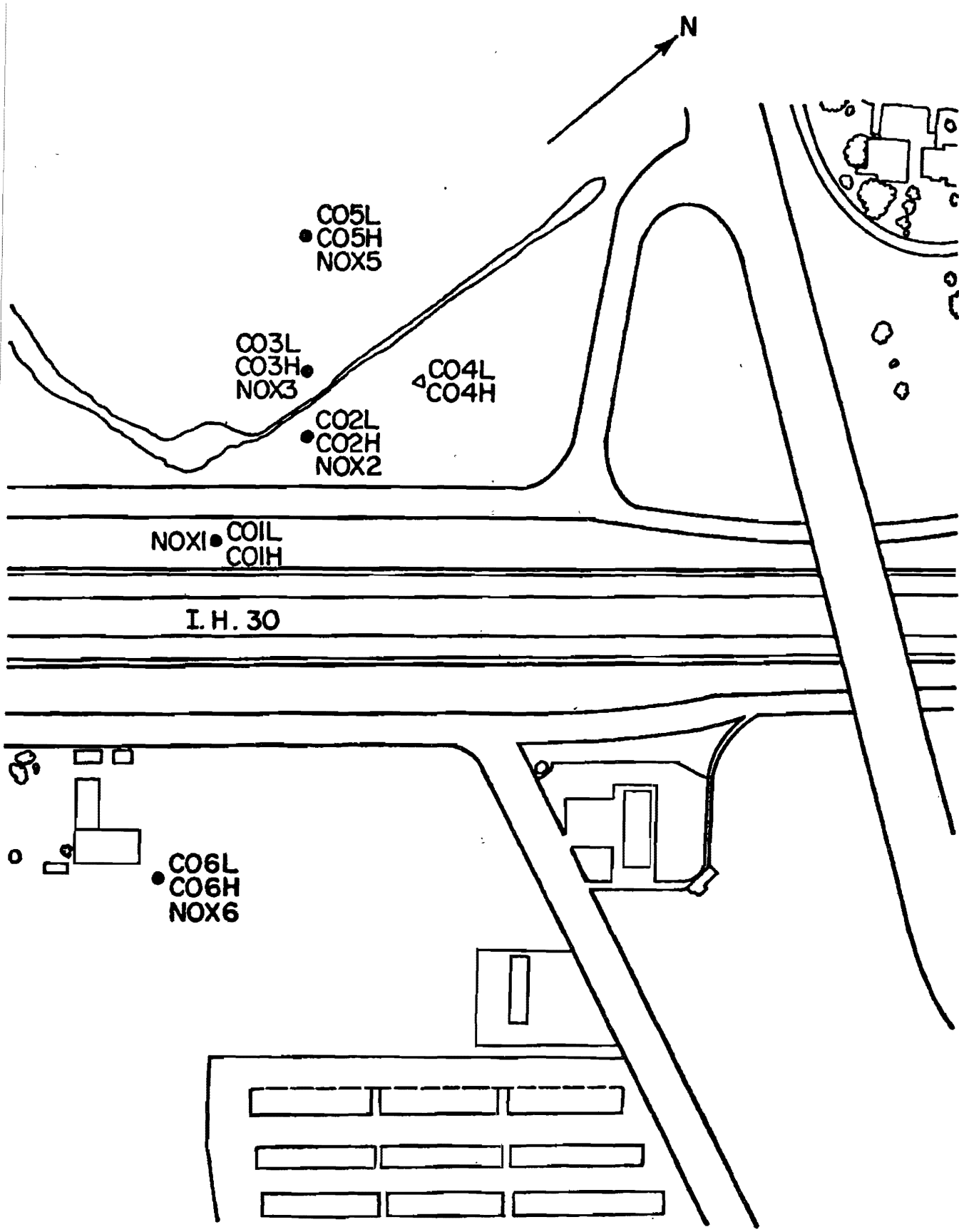


Figure 15

Overhead View of Dallas at-Grade Site

IH30 at Motley Dr

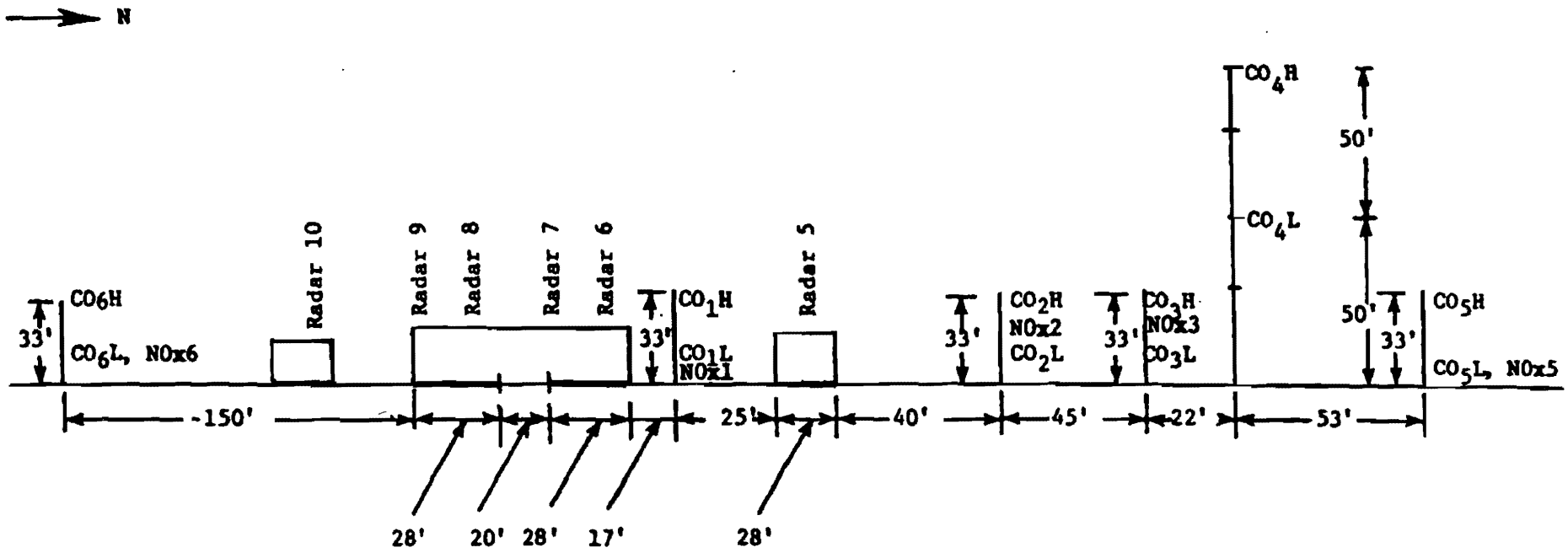


Figure 16

Cross Section of the Dallas at-Grade Site—IH30 at Motley Dr.

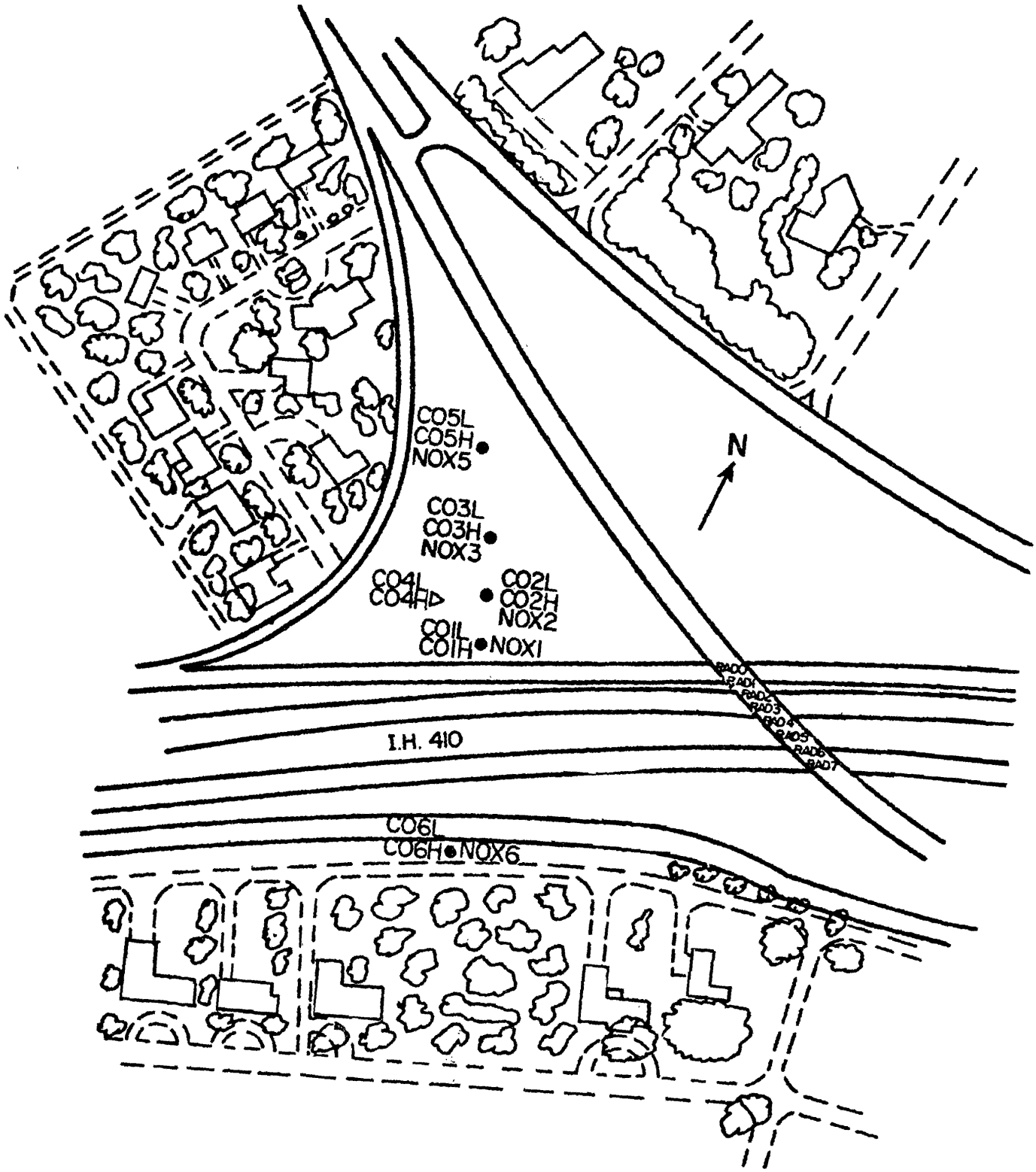


Figure 17
Overhead View of San Antonio Site
IH410 at Military Highway

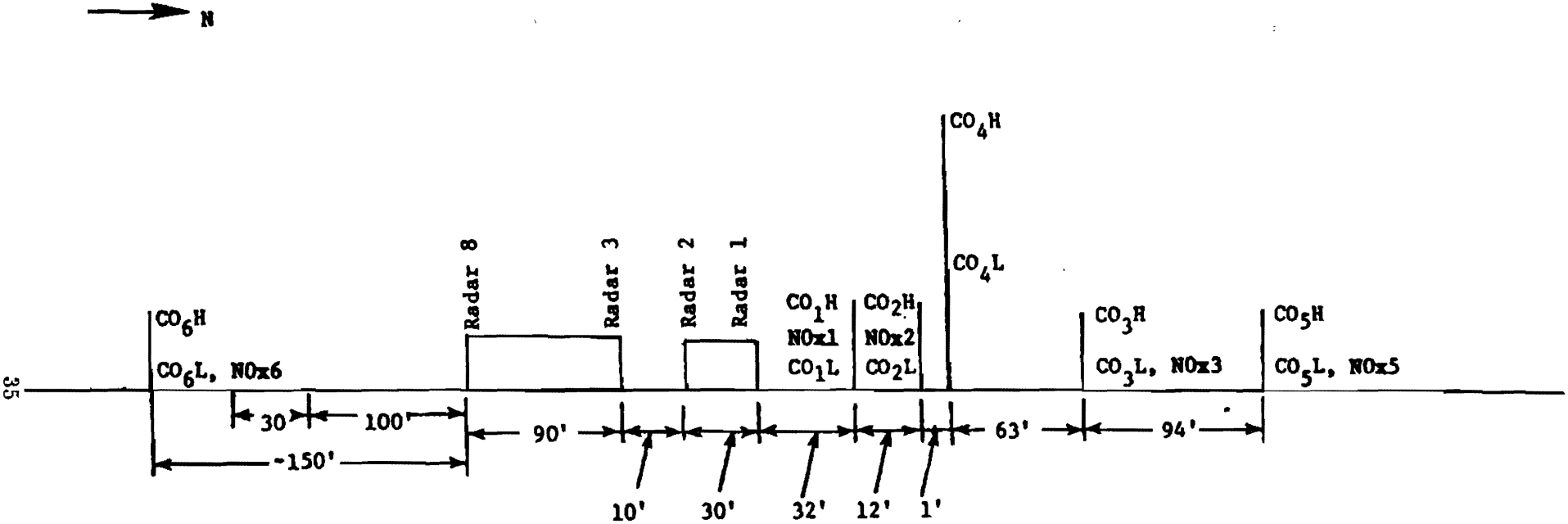


Figure 18

Cross Section of the San Antonio Site—IH410 at Military Hwy.

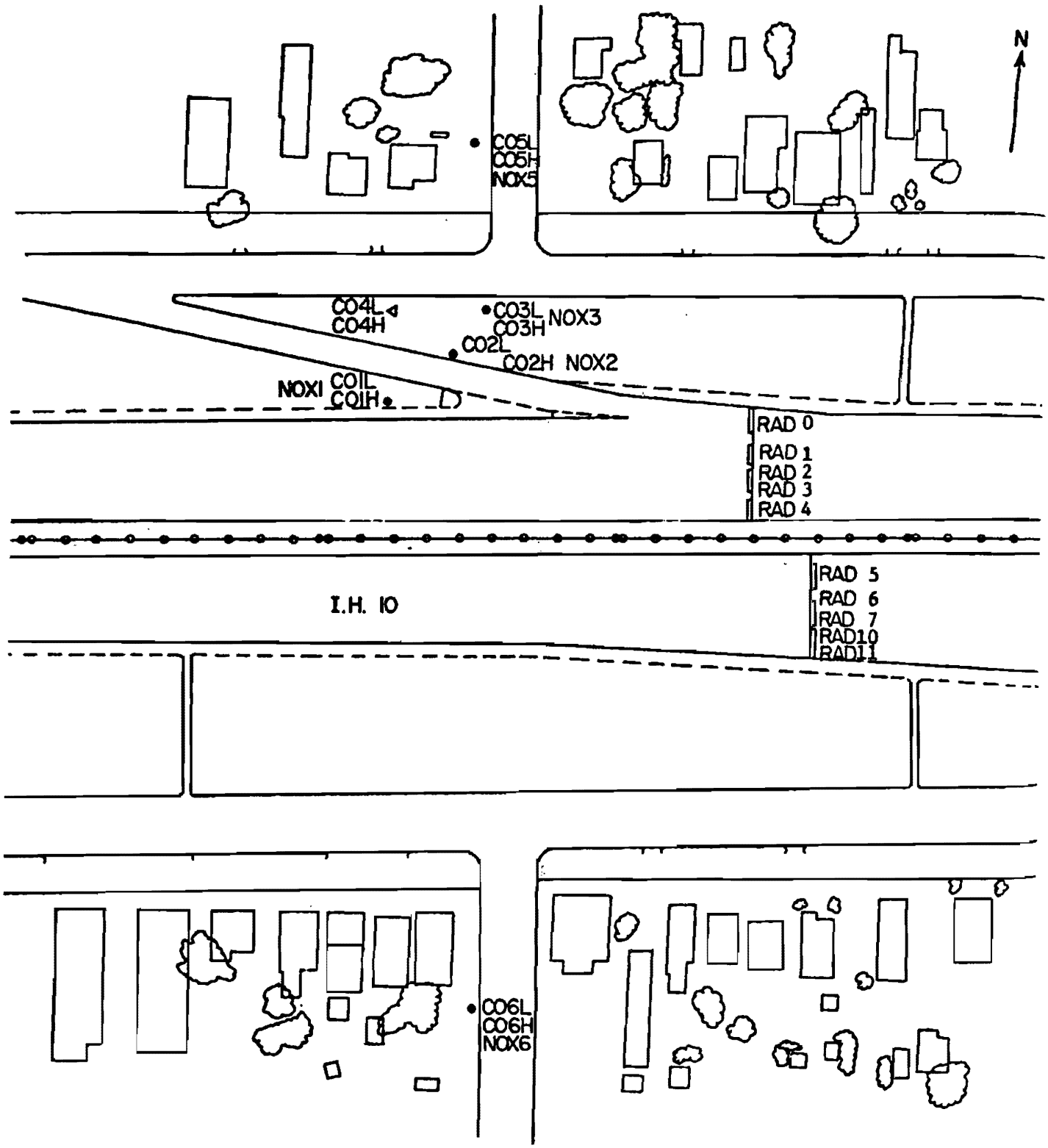


Figure 19

Overhead View of El Paso Site

IH10 at Luna St

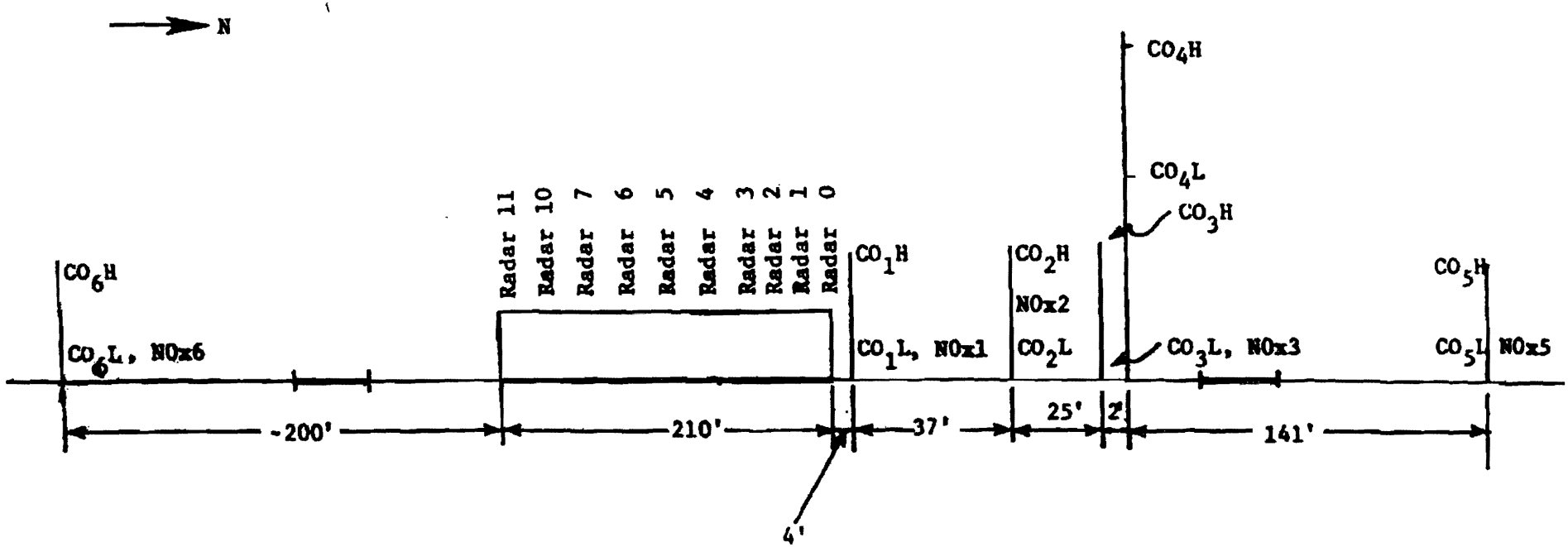


Figure 20

Cross Section of the El Paso Site—IH10 at Luna Street

Later studies at Texas A&M established data bases that were suitable for use in intersection model development. These studies were documented by Bullin, *et al.*[37] The research areas were located in Houston and College Station, Texas. Similar instrumentation was utilized in the collection of these data bases as was used in the earlier studies.

The College Station intersection research site was located at the corner of Texas Avenue, Jersey, and Kyle Streets. Figure 21 gives an overhead view of the site geometry. The terrain surrounding the research area is relatively flat. The northwest quadrant of the site contains a golf course with grass-covered ground and lightly scattered trees. The northeast and southeast quadrants contain single family residences. A small shopping center consisting of single-story buildings and an automobile service station is located in the southwest quadrant. Texas Avenue and Jersey Street were well traveled, while Kyle Street had a relatively low traffic volume. Towers 1, 2, and 3 were placed in the southeast quadrant 35 feet from Texas Avenue, and 35, 125, and 355 feet from Kyle Street, respectively. Tower 4 was in the southwest quadrant, 65 feet from Texas Avenue and 220 feet from Jersey Street. Tower 5 was 120 feet north and west of Texas and Jersey, located in the golf course.

Towers 1, 2, and 4 were used to also support air samplers for sulfur hexafluoride tracer gas and aerosol studies. The samplers were suspended at 5, 15, and 35 feet opposite corresponding meteorological stations.

The Houston intersection study was located at the corner of Woodway Boulevard and South Post Oak Lane, four blocks west of the West Loop (IH610). An overhead view of this *street canyon* site is presented in Figure 22 and an aerial photograph in Appendix J.

The northwest quadrant contains a service station at the intersection corner and two-story apartment buildings. A seven-story condominium occupies the northeast quadrant. The southeast quadrant contains three tall office buildings (one 18-story and two 24-story buildings). A 14-story condominium is located in the southwest quadrant.

Four towers were used at the Houston site with T1 being the southernmost and T4 the northernmost. T1 was in the southeast quadrant, 120 feet from Woodway and 20 feet back from South Post Oak. T2 was at the northeast corner, 10 feet from Woodway and South Post Oak. T3 and T4 were both positioned in the northwest quadrant, 10 feet west of South Post Oak with T3 95 feet from Woodway and T4 345 feet north. The mobile laboratory was parked just north of T3. Sampler intakes were located at 5, 20, and 35 foot heights on T2 and T3 and at 10 and 35 feet on T1.

Loop detectors were used to measure traffic flow at both the College Station and Houston intersection sites. At both sites, time lapse photographs were taken to verify the loop data. The reader who requires more detail on these intersection studies is referred to TTI report 250-2F.[37]

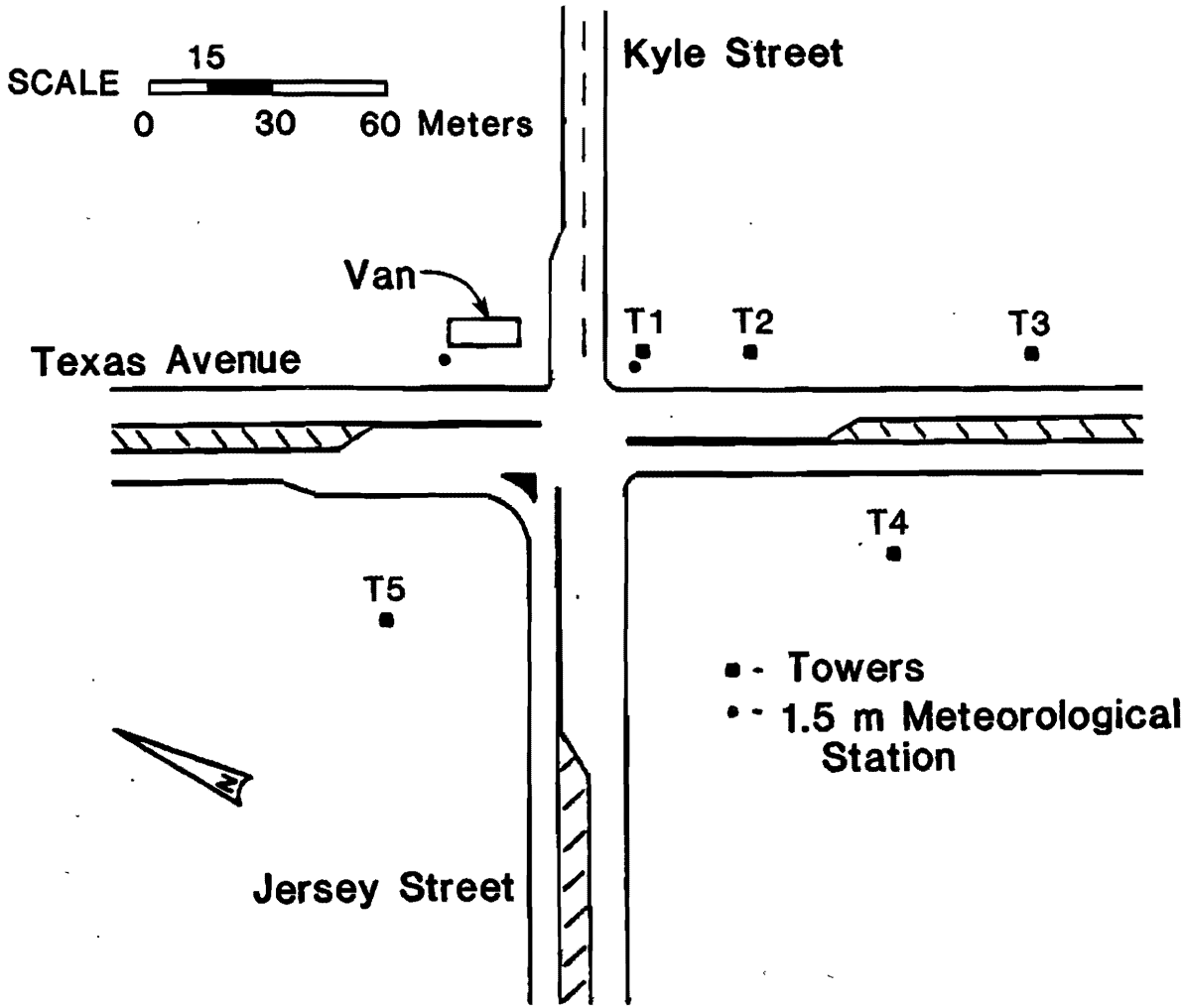


Figure 21

College Station Intersection Research Site

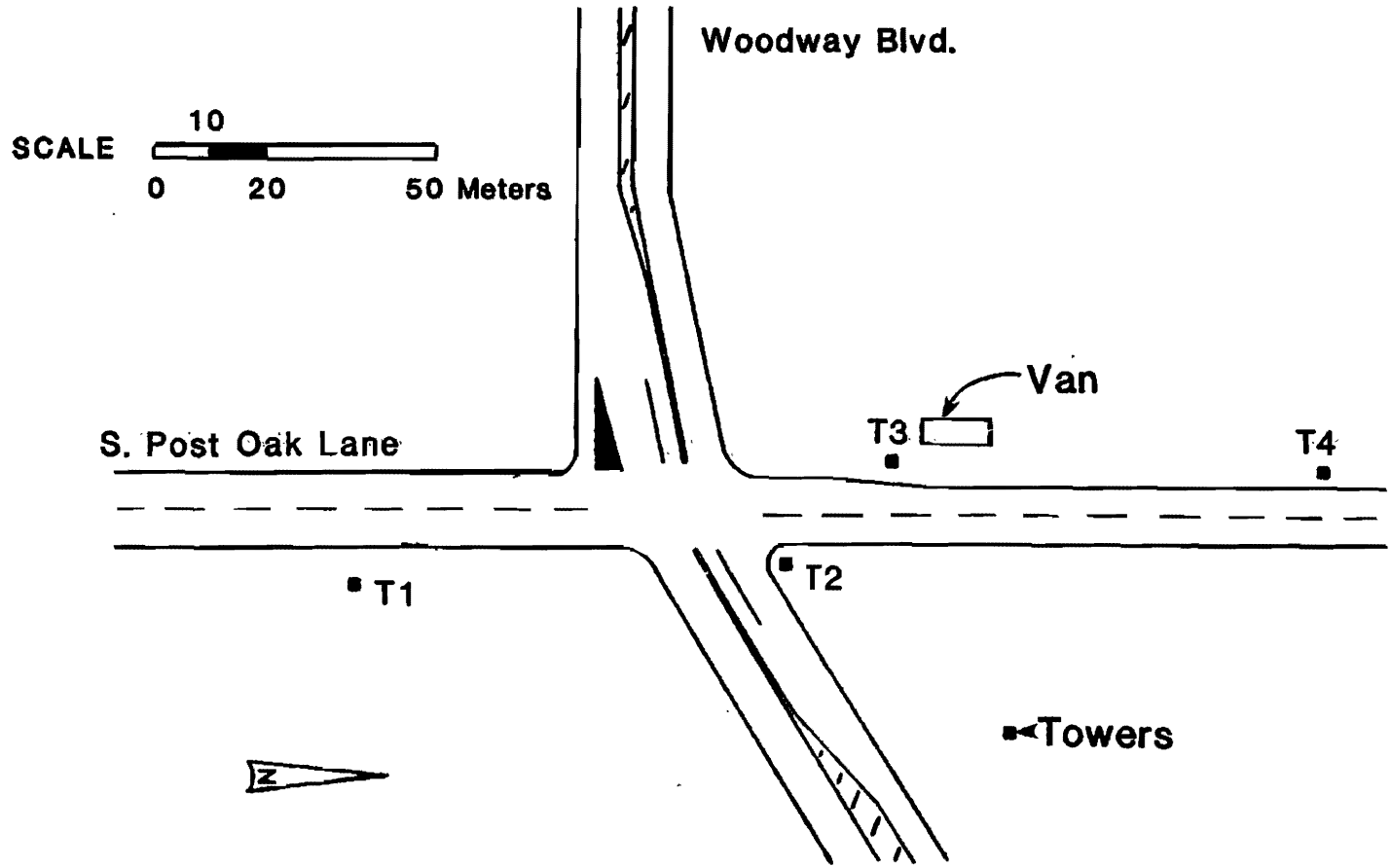


Figure 22

Houston Intersection Research Site

C. Stanford Research Institute Data Base

Dabberdt, *et al.*[38] give the results of an extensive experimental project performed by the Stanford Research Institute (SRI). Dispersion experiments were performed at ground-level, elevated, and depressed roadway sections. Two different tracer gases were used as well as several wind tunnel studies.

The at-grade experiment was conducted in the San Francisco Bay area on a stretch of U. S. Highway 101, in Santa Clara, California. The road is a major freeway, with three lanes of traffic in each direction. The surrounding area consists primarily of single level homes.

During the data collection periods, two vans were driven continuously in the traffic stream. The vehicles always drove in the center lane at the average traffic speed. SF_6 was released in the westerly direction while fluorotribromomethane was released in the easterly direction. Both tracers were emitted at a uniform rate between points approximately 400 meters to either side of the sampling line.

The side view of the at-grade site is shown in Figure 23. Traffic volumes were estimated using sensor cables laid across the roadway. Comprehensive traffic information including speed and axle number for each vehicle was recorded. UVW anemometers and propeller vanes determined all meteorological data every 2.5 seconds. Records of temperature and insolation were also made as in the Texas A&M data base.

Continuous concentrations of material were not recorded. Hourly air samples were obtained by sequential multiple bag samplers. The bags were made of clear Tedlar and had a volume of approximately five liters. A modified Perkin-Elmer gas chromatograph determined the tracer gas concentration while a Beckman Model B6800 Air Quality Chromatograph measured concentrations of carbon monoxide, methane, and hydrocarbons.

The elevated diffusion experiment was conducted at a viaduct section of Interstate 280 in San Jose, California. This section consisted of two 7 meter high viaducts, each about 24 meters wide. The viaducts were separated by 15 meter gaps. The top of the viaducts were just above the roof level of several two-story houses located on each side of the roadway. The experiments conducted at this site were performed in a manner similar to those at the at-grade site. Instrumentation is shown in Figure 24.

The cut section experiments are not included in this report. The receptors were located inside a deep-cut section and could not be easily modeled.

D. CALTRANS Sacramento Intersection Study

The California Department of Transportation collected pollutant, traffic, and meteorological data at the intersection of Florin Road and Freeport Boulevard in Sacramento, California, during the months of February, March, and April, 1981.[39] Continuous measurements were made at the

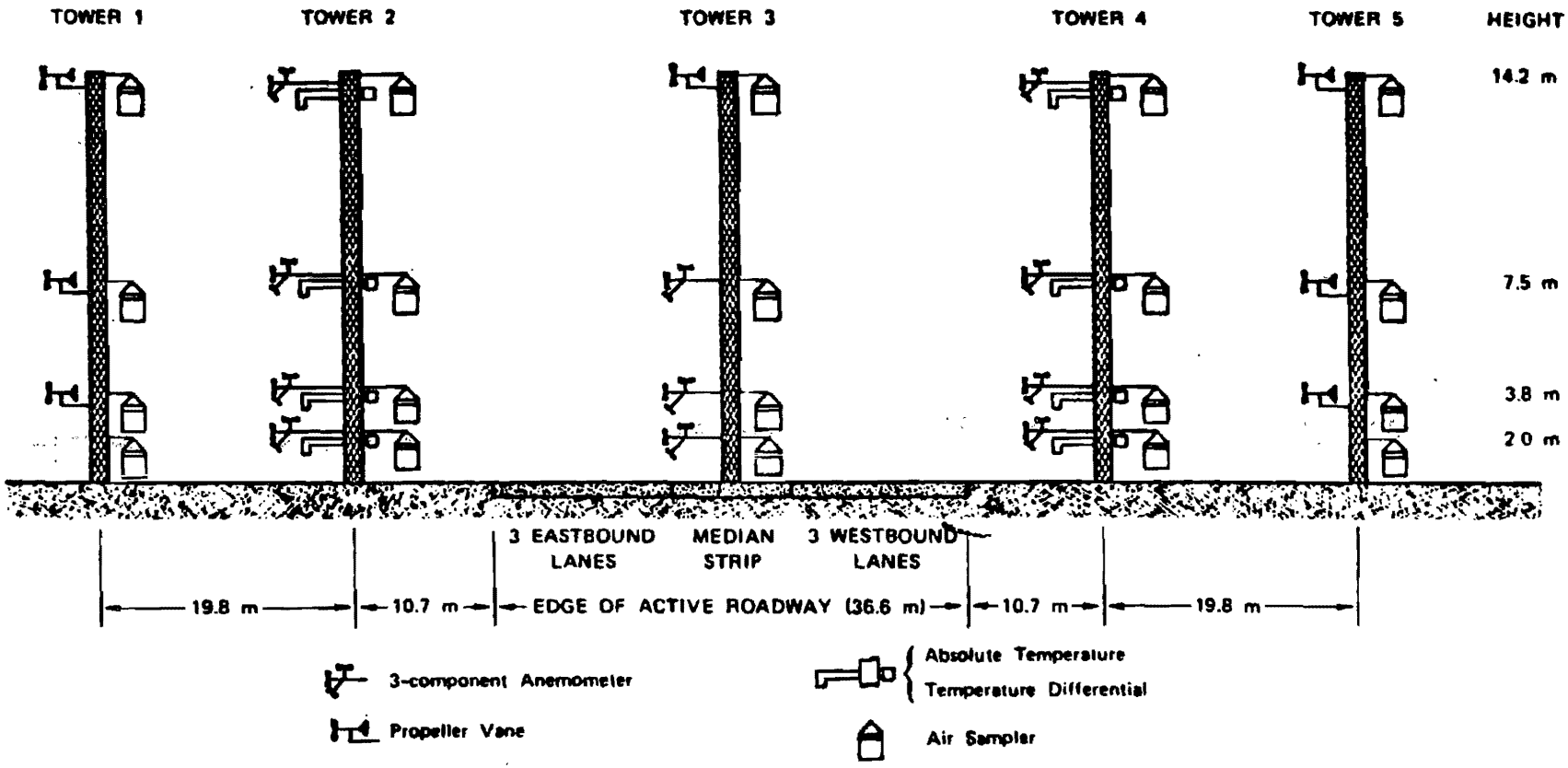


Figure 23

Stanford Research Institute at-Grade Site[38]

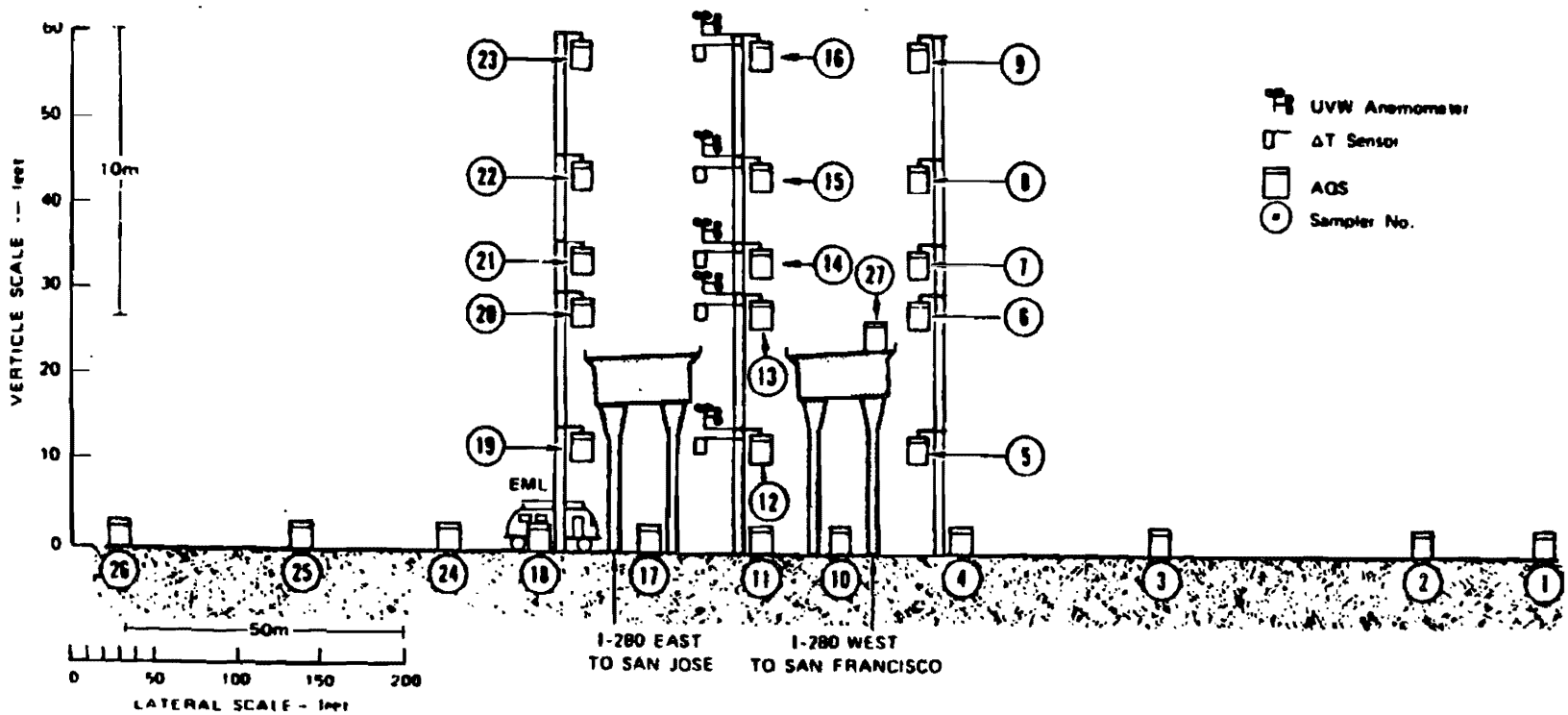


Figure 24

Stanford Research Institute Elevated Site[38]

site for forty days. The site and instrument layouts are presented in Figure 25.

The terrain consists of bare or grass covered ground for a distance of at least 50 meters from the roadway in all four quadrants. The land is level and is occupied by scattered single story residential developments. A small shopping center is located well back from the intersection in the northwest quadrant. The site offers reasonably high traffic flow without interfering background sources such as parking lots.

Fifteen carbon monoxide probe locations were utilized. Eight of these were in the northwest quadrant and seven in the southwest quadrant. A sequential bag sampler was also placed in the southwest quadrant. Two systems were used to monitor carbon monoxide concentrations: non-dispersive infrared (NDIR) analyzers and gas chromatographs with flame ionization detectors (FID). The two sample towers nearest to Florin Road contained the sample probes, with four probes on the south tower at 1, 2, 4, and 10 meter heights, and five probes on the north tower at 1, 2, 4, 10, and 15 meter levels. Three additional probes were placed in both the northwest and southwest quadrants at a height of one meter. Three NDIR detectors were used to obtain concentration profiles. Each analyzer was coupled to five probe sample lines and a microcomputer performed valve switching at one minute intervals so that each sample line was analyzed for one minute out of a five minute period. The gas chromatography analyses were run only for the nine probes at the two towers innermost to Florin Road.

The outermost meteorological towers had cup anemometers and temperature probes at the 2 and 10 meter levels, and wind vanes at the 10 meter level. Bivane anemometers were also mounted at a height of four meters on the innermost towers. Pneumatic counters obtained traffic counts. No data concerning the percentage of vehicles turning or vehicle speeds were recorded.

The CALTRANS data base consists of hourly averages for the instruments mentioned above. Additionally, hourly averages for the Richardson number were calculated.

E. Other Data Bases

Many other data bases have been established that are suitable for roadway pollutant dispersion modeling. A comprehensive review of data sets collected in several states was presented by Green.[21] These data sets included the following contributors: North Carolina, by Noll[40]; Tennessee, by Noll, *et al.*[41]; Virginia, by Carpenter and Clemeña[42]; Illinois, by Habegger, *et al.*[43]; California, by Ranzieri, Bemis and Shirley[44]; and Washington, by Badgely, *et al.*[45] Another important study not mentioned by Green was conducted in New York State by Rao, *et al.*[46]

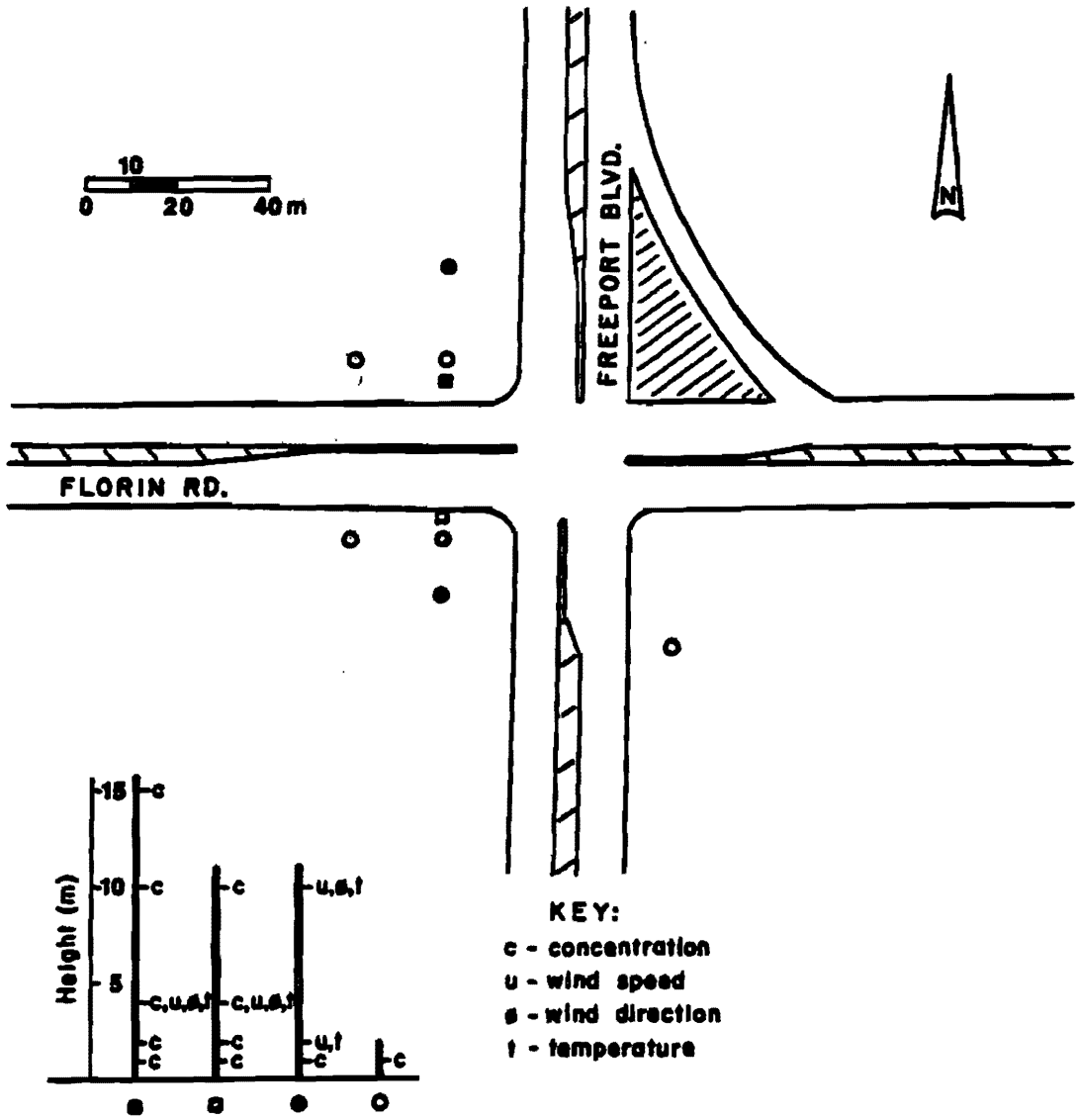


Figure 25

CALTRANS Intersection Study[39]

Chapter 3

Model Development

The scope of this project was divided into two main segments. The first of these included the revision and improvement of two previously developed pollution dispersion models. The second goal was the collection of a large data base that could be used for further model verification or development. The first of these goals, model improvement and revision, is discussed in this chapter.

I. TXLINE-2

TXLINE-2 is a second generation computer model used to predict pollutant concentrations downwind of a singular finite line source or several parallel finite line sources at any elevation. The revised model was developed by Schroeder[47] in conjunction with this project. The original TXLINE model was written by Rodden and is discussed by Bullin, *et al.*[2]

TXLINE-2 is primarily intended for predicting carbon monoxide concentrations, but also may be used to simulate the dispersion of other gaseous pollutants. Since TXLINE-2 is a microscale model, it is not applicable for predicting concentrations at great distances from the initial sources.

There are several major differences between TXLINE-2 and other dispersion models currently in use. TXLINE-2 is the only current model that does not assume a flat wind profile. Instead, the model assumes a power law wind speed profile. Unlike models such as CALINE4 and HIWAY-2, TXLINE-2 does not use the simple Gaussian solution to the diffusion equation. Therefore, the uncertainties that arise in estimating the statistical Gaussian dispersion parameters are avoided.

One additional feature of TXLINE-2 is a low wind speed correction factor. Comparison of TXLINE-2 to both GM data[35] and Texas A&M data[11] demonstrated that TXLINE-2 overpredicted concentrations as wind speed and/or angle were decreased. The wind speed correction factor was fit through analysis of the GM data to better represent those cases. This correction factor approaches unity as the wind speed approaches 4 meters/second and as the wind angle approaches 0 degrees (perpendicular to the road). This factor is recommended for all dispersion modeling; however, the user is given the option to omit the factor during the calculational procedure.

A. Development of Finite Length (Link) Capabilities

The original TXLINE model represented the roadway as a line source of infinite length. This procedure limited the use of the original program to long, straight sections of roadways with no complicating factors such as intersections. A finite length version of the model would have the added capabilities of modeling short sections of roadways, curves, and intersections. Additionally, such a model would be capable of modeling dispersion upwind of the roadway.

There were three major problems encountered in developing a finite line version of the model. First, the roadway had to be mapped onto its own coordinate system. Secondly, the coordinate

system had to be transformed into a form usable in the dispersion equations. Finally, the starting and ending points for numerical integration had to be determined.

The roadway was set on a map grid oriented with the $y-$axis running along the north-south line as shown in Figure 26. The endpoints of each link as well as each receptor location were assigned coordinates on the map grid. The wind angles were configured so that a westerly wind had an angle of 0°, a southerly wind had an angle of 90°, *etc.* It is possible to place the x and $y-$axes in another orientation, but the wind angles would have to be modified accordingly.

Once the roadway (or roadways) has been placed on a map grid, each link must undergo a coordinate transformation so that the link can be utilized by the dispersion equations. First, the bearing of the link with respect to the $x-$axis is determined by the following equations:

$$XD = XLIN2 - XLIN1 \tag{3-1}$$

$$YD = YLIN2 - YLIN1 \tag{3-2}$$

$$LLEN = \sqrt{XD^2 + YD^2} \tag{3-3}$$

$$HYP = |XD/LLEN| \tag{3-4}$$

$$LB = \arccos(HYP) \tag{3-5}$$

where:

$XLIN =$ $x-$coordinate of link endpoint
$YLIN =$ $y-$coordinate of link endpoint
$XD =$ length in $x-$direction
$YD =$ length in $y-$direction
$LLEN =$ length of link
$HYP =$ cosine of the angle LB
$LB =$ bearing of the link with respect to the $x-$axis

The quadrant of the angle LB is determined by the signs of XD and YD. A coordinate transformation is then performed on the link. The transformation aligns the $y-$axis with the link such that one endpoint of the link is at the origin of the coordinate system. Finally, the wind angle with respect to the link is determined. This process is illustrated in Figure 27.

The original TXLINE model used the point on the roadway that contributes maximum concentration of pollutant to the receptor as the starting point for numerical integration. This starting point was found using Newton's method to solve for the zero of the first derivative (with respect to length) of the dispersion equation. The starting value for the iterative procedure was determined with a fitted function. The revised version uses only the approximate value resulting from the fitted function since the starting point does not always fall on the finite length of a roadway. The

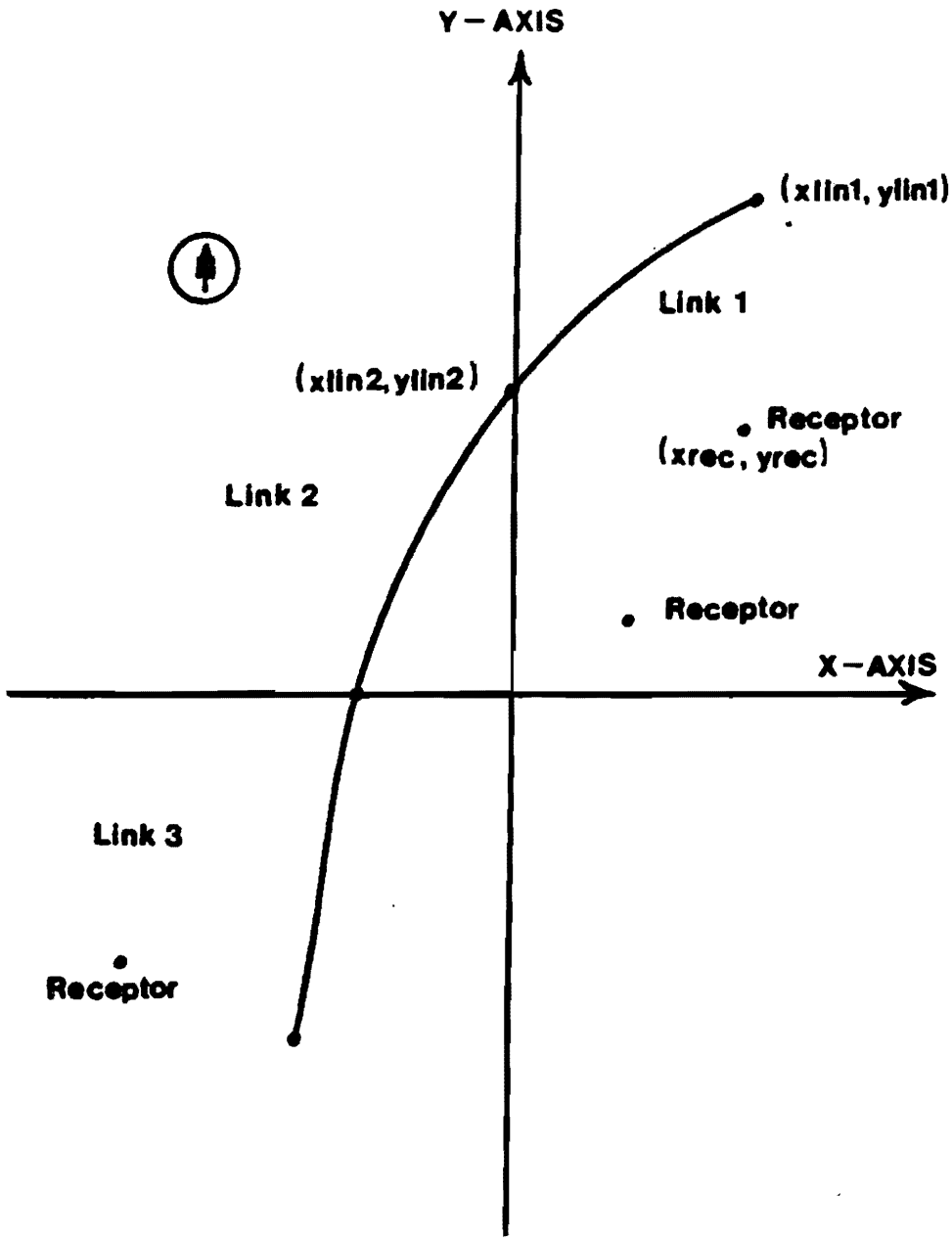


Figure 26

Mapping of the Roadway in TXLINE-2

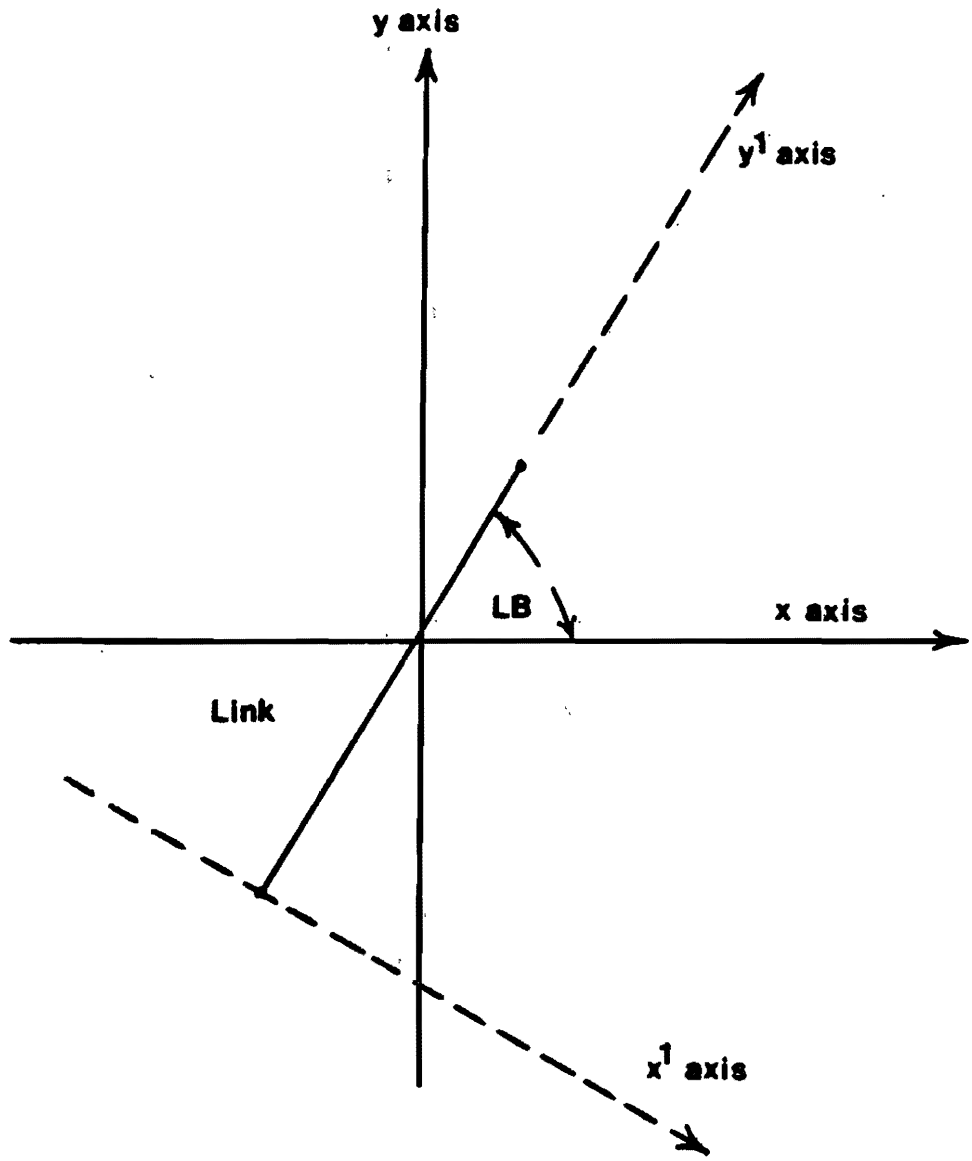


Figure 27

Link Coordinate Transformations in TXLINE-2

starting point is then compared to the endpoints of the link to insure that the point is on the link. If the starting point is not on the link, the point is placed on the link one segment away from the nearest end point.

A numerical integration of the point source equation along the roadway is then performed by TXLINE-2. Before each iteration, the program checks to see if the iteration will pass beyond the end of the link. If this occurs, the program modifies the segment length so that the iteration ends at the endpoint of the link. After the final iteration, the program terminates the integration. With these link additions, the model is capable of modeling dispersion upwind of the roadway.

B. Dispersion Equations Utilized by TXLINE-2

The dispersion equations used by the revised version of the model are essentially those presented in Chapter 2 for the original model (refer to equations 2-17 through 2-27). Modifications were made to the point source equation (2-19) by multiplying it by K_1/K_0 where,

$$K_0 = \frac{u_*^2 z_{ref}^{0.14285}}{0.14285 u_{ref}} \qquad (3-6)$$

where:

z_{ref} = reference wind speed height

u_{ref} = reference wind speed at z_{ref}

and altering the embedded parameter b (equation 2-20) to

$$b = (2+\mu)(1+2m) \qquad (3-7)$$

where:

μ = power law constant

A more detailed treatment of TXLINE-2 may be found in Schroeder.[47]

II. TEXIN2

TEXIN2 is the latest version of the original TEXIN intersection model developed by Nelli and described by Messina, *et al.*[5] The revised model was developed by Korpics[48] in conjunction with this project. The original model was developed to estimate carbon monoxide concentrations near signalized or unsignalized intersections, as well as several minor intersections arising from stop or yield-controlled side streets. The model performed the following distinct tasks: estimation of traffic parameters, estimation of vehicle emissions and their distribution, and modeling the dispersion process downwind of the roadway. The modification of these tasks is described in the following sections.

A. Estimation of Traffic Parameters

The first task performed by the model is a traffic flow analysis. Initially the traffic flow on the major intersection is evaluated and subsequently any minor intersections are handled. The process for determining the traffic parameters of unsignalized major and minor intersections has been retained from the original model, while an alternate method of evaluating signalized intersections has been added to TEXIN2.

The Critical Movement Analysis (CMA) Operations and Design Technique[49] was added to the new model for two to eight phase signalized intersections. The CMA Planning Procedure was retained from the original model as an option. The Operations and Design analysis treats the intersection as an entire unit, and is based on the principle that a combination of conflicting movements must be accommodated at each signalized intersection. The sum of these volumes represents the *critical volume.*

The critical volumes are the volumes of travel represented by the highest lane volume of opposing travel (left turn, through, and right turn) for both the north-south and east-west directions. The contributions are determined for each direction and added to give the sum of critical volumes. These critical volumes are compared to a benchmark intersection capacity to determine the Level of Service and volume to capacity ratio (V/C) for the intersection. The Level of Service is a measure of mobility of an intersection and is categorized as shown in Table 1. An average value of 1800 passenger cars per hour of green (pchg) for a twelve foot through traffic lane—with no trucks, buses, turns, or pedestrian interference—can be used as a base value for determining intersection capacity in the CMA technique.

The critical volumes for the north-south and east-west approaches are based on the traffic flow for each movement (left turns or through and right turns). Messer and Fambro[50] have established certain guidelines for combining these critical movements. Their results are shown in part of Table 2 and Figure 28. For multiphase traffic signals (those with three or more phases), the most probable phase sequence is first calculated from traffic flows adjusted for left turn signalization and other factors to passenger cars per hour (pch). The through plus right turn volume which moves through during a green arrow is then subtracted from the total through plus right turn volume and the remaining volume is carried over to the next phase.

A number of adjustment factors have been devised that affect traffic flow and hence modify critical volumes. These factors are:

(1) Left turns
(2) Bus and truck volume
(3) Peaking characteristics
(4) Lane width

Table 1

Intersection Levels of Service[49]

Level	Description
A	Free flow; low volume, high operating speed and high maneuverability
B	Stable flow, moderate volume; speed somewhat restricted by traffic conditions
C	Stable flow, high volume; speed and maneuverability determined by traffic conditions
D	Unstable flow, high volume; tolerable but fluctuating operating speeds and maneuverability
E	Unstable flow, high volume; approaching roadway capacity; limited speed, intermittent vehicle queueing
F	Forced flow; volume lower than capacity due to very low speeds; heavy queueing of vehicles, frequent stoppages

Table 2

Combining Critical Movement, Operation and Design Applications[49]

Signal Phasing and Intersection Geometry	Approaches[a]	Critical Movement
One phase, no left turn bay	1 and 2	A1B2 or A2B1
	3 and 4	A3B4 or A4B3
One phase, with left turn bay	1 and 2	A1 or A2 or B1 or B2
	3 and 4	A3 or A4 or B3 or B4
Two phases, no overlap, with left turn bay		
1. Leading or lagging left turns, from both directions	1 and 2	A1 or A2 + B1 or B2[b]
	3 and 4	A3 or A4 + B3 or B4[b]
2. Leading or lagging left turns, from one direction	1 and 2	B1 + A1 or A2[c]
	3 and 4	B3 + A3 or A4[c]
Two phases, with overlap, with left turn bay		
1. Leading or laggin left turns, from both directions	1 and 2	A1 + B1 or A2 + B2
	3 and 4	A3 + B3 or A4 + B4
2. Leading or lagging left turns, from one direction	1 and 2	B1 + A1 or A2[c]
	3 and 4	B3 + A3 or A4[c]

a. See Figure 28 for an identification of intersection movements and approaches.
b. Note that the critical volume on a given street is the single highest volume.
c. Assume arrow is for movements B1 and B3.

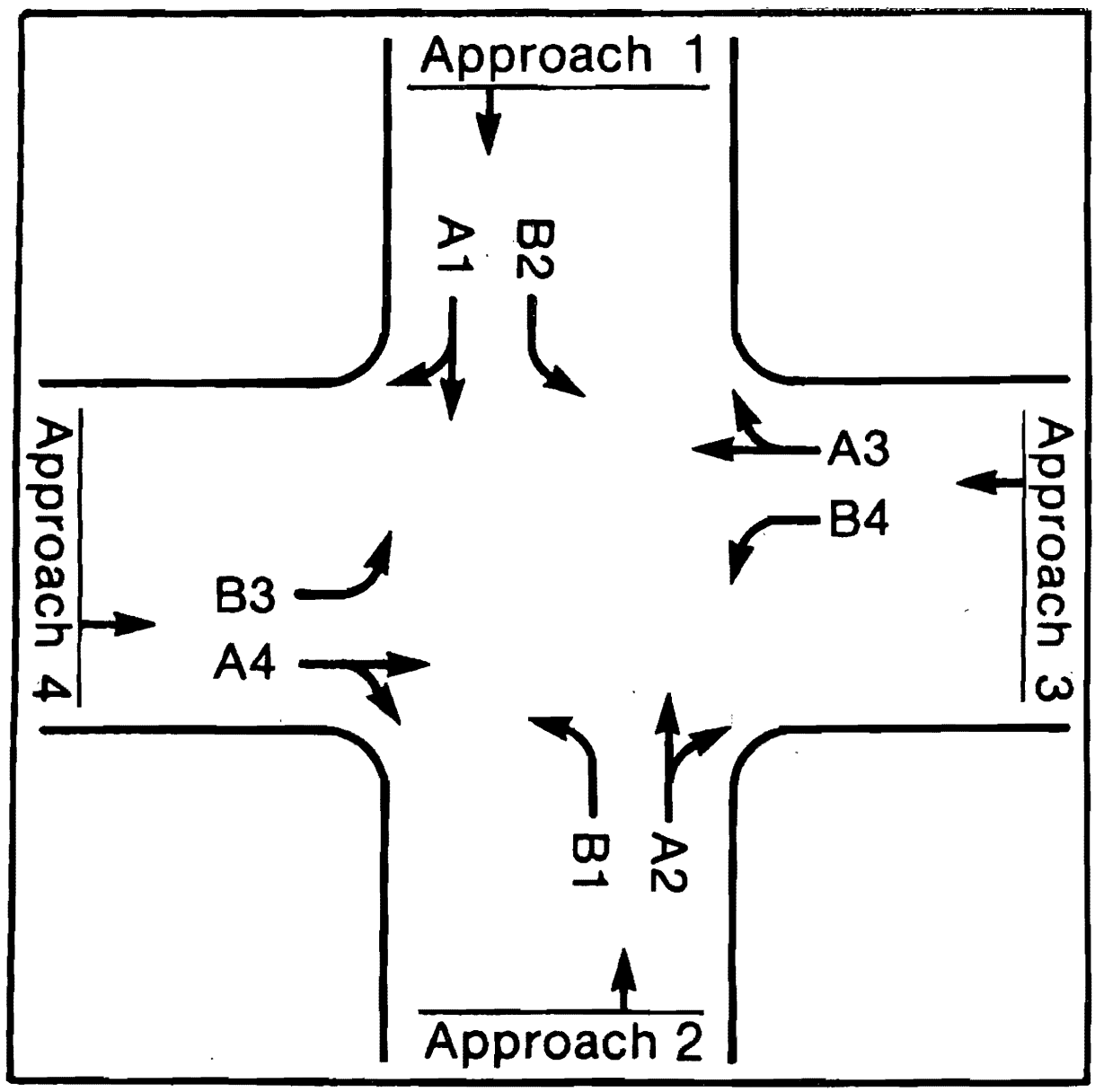


Figure 28

Intersection Movements for the CMA

Operations and Design Procedure[49]

(5) Bus stop operations

(6) Right turns with pedestrian activity

(7) Parking activity

The original model utilized only the first of these while the revised version allows for use of the first four. In both models, left turns are treated in detail because of their large impact on intersection capacity. This effect is created using passenger car equivalency (PCE) values. PCE values are multiplicative adjustment factors applied to left turning traffic. Table 3 gives PCE values for left turns on both left through and exclusive left turn lanes. Note that the opposing volumes needed to determine the PCE values are in vehicles per hour (vph) and not adjusted to passenger cars per hour (pch).

The additional factors employed in TEXIN2 are bus and truck volume, peaking characteristics and lane width. Trucks and buses reduce intersection capacity because their time headway is greater than the 2.0 second average implied by a set capacity of 1800 pchg. This is compensated for by counting a truck or bus as being equivalent to two passenger cars as recommended in the National Cooperative Highway Research Program (NCHRP) 3-28[49] report.

Peak five minute flow rates are converted to one hour volumes by the use of a peak hour factor (PHF). A study of several locations noted in the NCHRP report 3-28 suggests an average value of PHF to be 0.85. This value is used in TEXIN2 to increase the passenger car volume by 15 percent. Lane width adjustments are also utilized to alter the passenger car volume at intersections. Table 4 lists the recommended width adjustment factors used in TEXIN2. An emphasis should be on the fact that both of these factors are used to increase passenger car volume of the intersection, not reduce the capacity of the intersection.

Critical Movement Analysis is based on per lane volumes. To attain these per lane volume flows, TEXIN2 adjusts total volume flow on each intersection leg by use of a lane-use factor that converts total directional movement into a lane volume. Table 5 presents these factors. To account for unequal distribution of travel between lanes, the lane-use factors exceed the inverse of the number of lanes.

As part of the CMA techniques, a set of guidelines of Levels of Service, V/C ratios, average delay values, and sums of critical volumes were published. Correlations between Levels of Service and critical volumes are presented in Table 6a for the Operations and Design Technique while the relationship between V/C ratios and delay values is given in Table 7. In the revised model, the stopped delay for any V/C ratio can be determined through linear interpolation. This stopped delay per vehicle is the basis for determining several other traffic parameters in TEXIN2.

The CMA Planning Procedure provides an alternative method of calculating the intersection V/C ratio. This procedure is not as stringent as the Operations and Design Technique, and hence,

Table 3

Passenger Car Equivalency Values for Left Turn Effects[49]

Left Turns Allowed from Left-Through Lanes†				
1. No Turn Phase				
Opposing volume (vph):	000–299	300–599	600–999	1000+
1 Left Turn equals:	1.0 PCE	2.0 PCE	4.0 PCE	6.0 PCE
2. With Turn Phase				
1 Left Turn equals:	1.2 PCE			

Left Turns Allowed from Left Turn Bays Only‡				
1. No Turn Phase				
Opposing volume (vph):	000–299	300–599	600–999	1000+
1 Left Turn equals:	1.0 PCE	2.0 PCE	4.0 PCE	6.0 PCE
2. With Turn Phase				
1 Left Turn equals:	1.05 PCE			

†These values are used in both the CMA Operations and Design Technique and the CMA Planning Procedure to compute the effects of left turns.

‡These values are used only in the CMA Operations and Design Technique to compute the effects of left turns.

Table 4

Lane Width Adjustment Factors[49]

Lane Width (ft)	Lane Width Adjustment Factor
8.0–9.9	1.10
10.0–12.9	1.00
13.0–15.9	0.90

Table 5

Lane-Use Factors[51]

Approach Lanes	Lane-Use Factor
1	1.00
2	0.55
3	0.40
4	0.30

Table 6a

Level of Service Ranges

CMA Operations and Design Technique[49]

Level of Service	Maximum Sum of Critical Volumes (pch) Two Phase	Three Phase	Four or More Phases
A	1000	950	900
B	1200	1140	1080
C	1400	1340	1270
D	1600	1530	1460
E	1800	1720	1650
F		— Not Applicable —	

Table 6b

Level of Service Ranges

CMA Planning Procedure[49]

Level of Service	Maximum Sum of Critical Volumes (vph) Two Phase	Three Phase	Four or More Phases
A	900	855	825
B	1050	1000	965
C	1200	1140	1100
D	1350	1275	1225
E	1500	1425	1375
F		— Not Applicable —	

Table 7

Delay and Level of Service[49]

Level of Service	Typical V/C Ratio†	Delay Range‡ (sec/veh)
A	0.00–0.60	0.0–16.0
B	0.61–0.70	16.1–22.0
C	0.71–0.80	22.1–28.0
D	0.81–0.90	28.1–35.0
E	0.91–1.00	35.1–40.0
F	varies	40.1 or more

†Volume to capacity ratio

‡Measured as *stopped delay* as described in reference 53. Delay values relate to the mean stopped delay incurred by all vehicles entering the intersection. Note that traffic signal coordination effects are not considered and could drastically alter the delay range for a given V/C ratio.

normally calculates a smaller V/C ratio. The major differences between the CMA Operations and Design Technique and the CMA Planning Procedure include the use of vehicles per hour (vph) instead of passenger cars per hour (pch) to calculate the V/C ratio, the use of different combining techniques in calculating the sum of critical volumes, and the absence of all of the adjustment factors present in the Operations and Design Technique except those for left turns.

As indicated in Table 3, no left turn adjustment factors are applied when left turns are made from exclusive left turn bays in the CMA Planning Procedure. To determine the critical volumes for each signal phase, the highest conflicting traffic volume (on a per lane basis) is determined. Therefore, for a two phase signal, the critical volume is the largest total of through plus right turns added to the opposing left turn volume for the phase. For multiphase signals, the procedure is similar to that of the Operations and Design Technique. A major difference between the Operations and Design Technique and the Planning Procedure is that combining of through and opposing left turns is not done in Operations and Design applications. Once the critical volumes for both the north-south and east-west directions are determined, the sum of critical volumes (in vph) is computed and compared to the benchmark values listed in Table 6b to determine the V/C ratio as in the Operations and Design Technique.

Another change made in TEXIN2 deals with V/C ratios greater than 1.00. This condition represents breakdown conditions (Level of Service F) under which the CMA is not completely applicable and cannot accurately describe traffic flow conditions. The original version linearly extrapolated the stopped delay value beyond the normal V/C range of 0.00 to 1.00. Since it is reasonable to assume that any given vehicle will remain at the intersection for at least one complete cycle under these conditions, the minimum delay time which should be predicted is one cycle length as suggested by Parikh.[52] Table 8 lists the values suggested by Parikh for different degrees of intersection breakdown.

Once the stopped delay per vehicle ($SDPV$) is calculated from the V/C ratio and Table 7, the percent of vehicles stopping, $PCST$ is determined:[53]

$$PCST = 54.97 \times \log_{10}(ADPV) - 14.04 \qquad (3-8)$$

where

$ADPV$ = approach delay per vehicle (sec) and is given by:

$$ADPV = 1.316 \times SDPV + 1.303 \qquad (3-9)$$

Finally, the total queue length, QL, is determined for signalized intersections as the sum of the individual queue lengths for all approach legs by the equation:

$$QL = \frac{PCST \cdot TTEI \cdot 8 \cdot CY}{3600} \qquad (3-10)$$

Table 8

Stopped Delay per Vehicle under Breakdown Conditions[52]

Volume to Capacity Ratio	Stopped Delay per Vehicle (sec)
$1.0 < V/C \leq 1.05$	$1.0 \times CY^{\dagger}$
$1.05 < V/C \leq 1.10$	$1.2 \times CY$
$V/C > 1.10$	$1.5 \times CY$

†CY is the cycle length in seconds

where:

$PCST$ = the percent of vehicles stopping

$TTEI$ = the total number of vehicles entering the intersection on a per lane basis (veh/hr)

CY = cycle time (sec)

8 = distance represented by a queued vehicle (m)

TEXIN2 sets the minimum queue length for each leg of the intersection equal to one car length (8 m).

The methodology for unsignalized intersections is somewhat different from that for signalized intersections. The procedure, adapted from the NCHRP 3-28 report, is applicable only to intersections controlled by two-way stop signs or yield signs, and remains virtually unchanged from the original model. The algorithm is based on potential capacities for the minor approach movements which are compared to the existing demand for each movement to determine the Level of Service.

First, all movements on the minor street and the left turns from the major street are corrected for the vehicle mix to give passenger car equivalencies per hour. The factors used are 0.5, 1.0, 1.5, and 2.0 for motorcycles, passenger cars, trucks, and truck-trailers, respectively. Next, conflicting traffic streams, M_H, as described in Figure 29, are used to determine a maximum capacity, M_{NO}, for a given movement. For vehicles emerging from the minor road (or turning left off the major road), the available gaps in the conflicting streams must be long enough to accommodate the desired maneuver. Table 9 illustrates critical gaps dependent upon the intended maneuver, the type of control, the prevailing speed on the major road, and the number of lanes on the major road. The maximum capacity is determined from Figure 30 and information on conflicting volume and critical gaps. This maximum capacity is adjusted for congestion interference by an impedance factor, P, which defines the probability that the minor road movement will remain unaffected by traffic flow from the major road to the minor road.

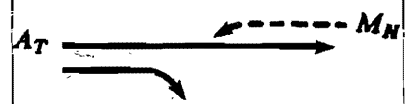
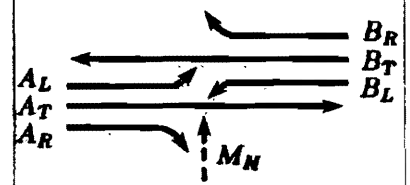
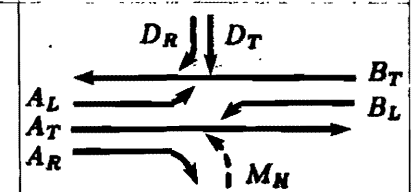
Figures 31 and 32 illustrate the impedance factor as a function of the percent capacity used and the manner by which maximum capacities for each movement are reduced, respectively. A final adjustment is needed to account for shared lane conditions whereby an exclusive lane does not exist for a particular movement. The resulting interference between potential movements reduces the capacity of the lane. This can be modeled by the following equation extracted from the NCHRP 3-28 report:

$$\frac{1}{M_{134}} = \frac{X}{M_1} + \frac{Y}{M_3} + \frac{Z}{M_4} \qquad (3-11)$$

where:

M_{134} = capacity of all streams using the shared lane

X, Y, Z = proportion of right, through, and left movements, respectively

Step 1	Right turns into major street $M_H = \frac{1}{2}A_R + A_T$	
Step 2	Left turns from major street $M_H = A_R + A_T$	
Step 3	Crossing Major street $M_H = \frac{1}{2}A_R + A_T + A_L + B_L + B_T + B_R$	
Step 4	Left turns into major street $M_H = \frac{1}{2}A_R + A_T + A_L + B_L + B_T + D_T + D_R$	

Notes:

In Step 1, if there is more than one lane on the major street, A_T is the flow in the curb lane only.

In Steps 1, 3, and 4, if a turning lane is present for major street right turns, A_R can be omitted.

In Steps 2 and 3, large radius turning areas for right turns off the major street and/or STOP or YIELD control of these turns reduce or eliminate the effect of A_R or B_R.

For complementary movements, reverse the major street movements (A and B) and minor street movements (C and D).

Figure 29

Unsignalized Intersection Conflicting Traffic Schemes[49]

Table 9

Critical Gaps (sec) for Passenger Cars[49]

Vehicle Maneuver and Type of Control	Prevailing Speed: 30 mph (50 kph) Major Road, 2 Lanes	30 mph (50 kph) Major Road, 4 Lanes	55 mph (90 kph) Major Road, 2 Lanes	55 mph (90 kph) Major Road, 4 Lanes
Right Turn from Minor Road:				
Yield Control	5.0	5.0	6.0	6.0
Stop Control	6.0	6.0	7.0	7.0
Left Turn from Major Road:				
No Control	5.0	5.5	5.5	6.0
Crossing Major Road:				
Yield Control	6.0	6.5	7.0	8.0
Stop Control	7.0	7.5	8.0	9.0
Left Turn from Minor Road:				
Yield Control	6.5	7.0	8.0	9.0
Stop Control	7.5	8.0	9.0	10.0

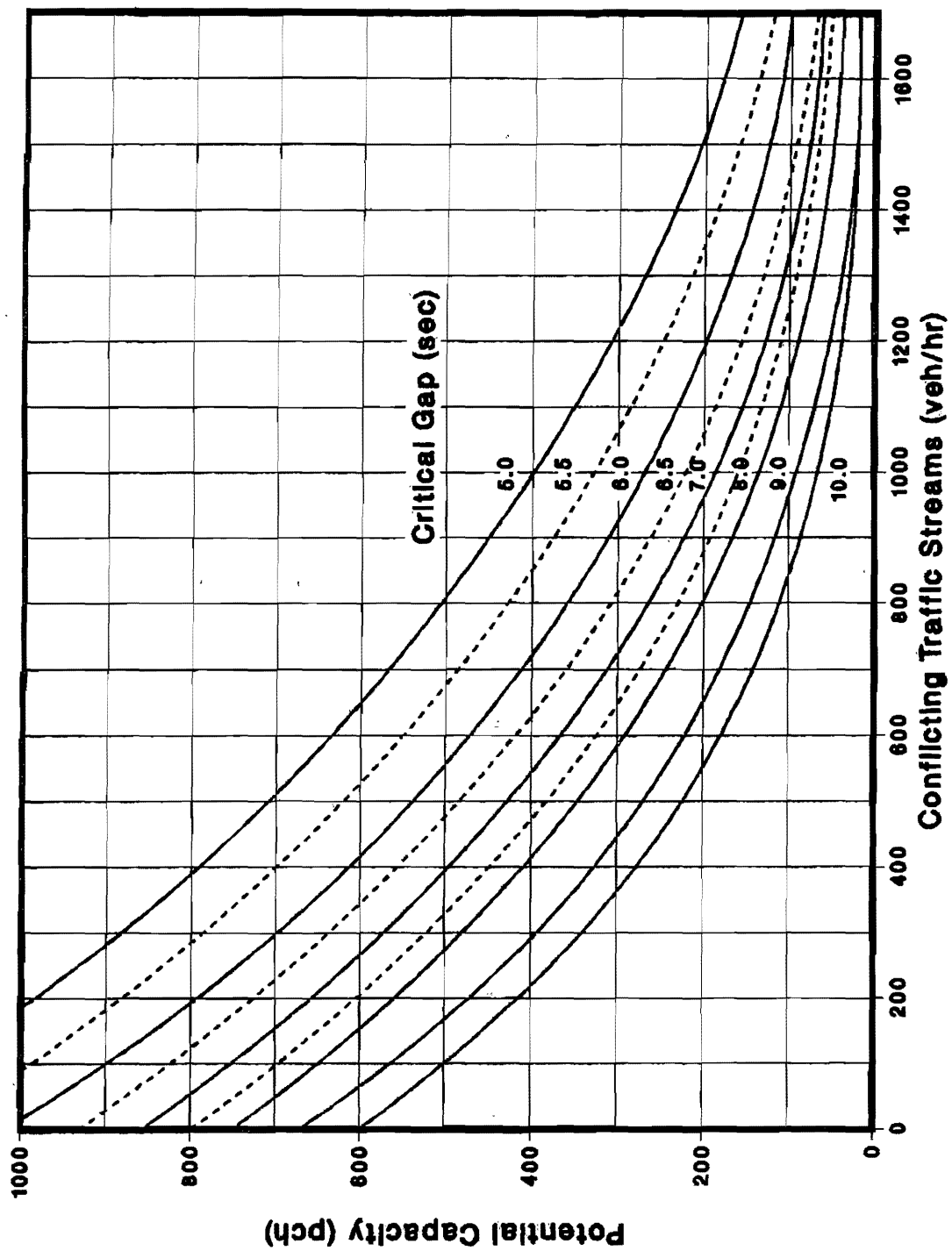


Figure 30

Maximum Capacity Based on Conflicting Volume and Critical Gap[49]

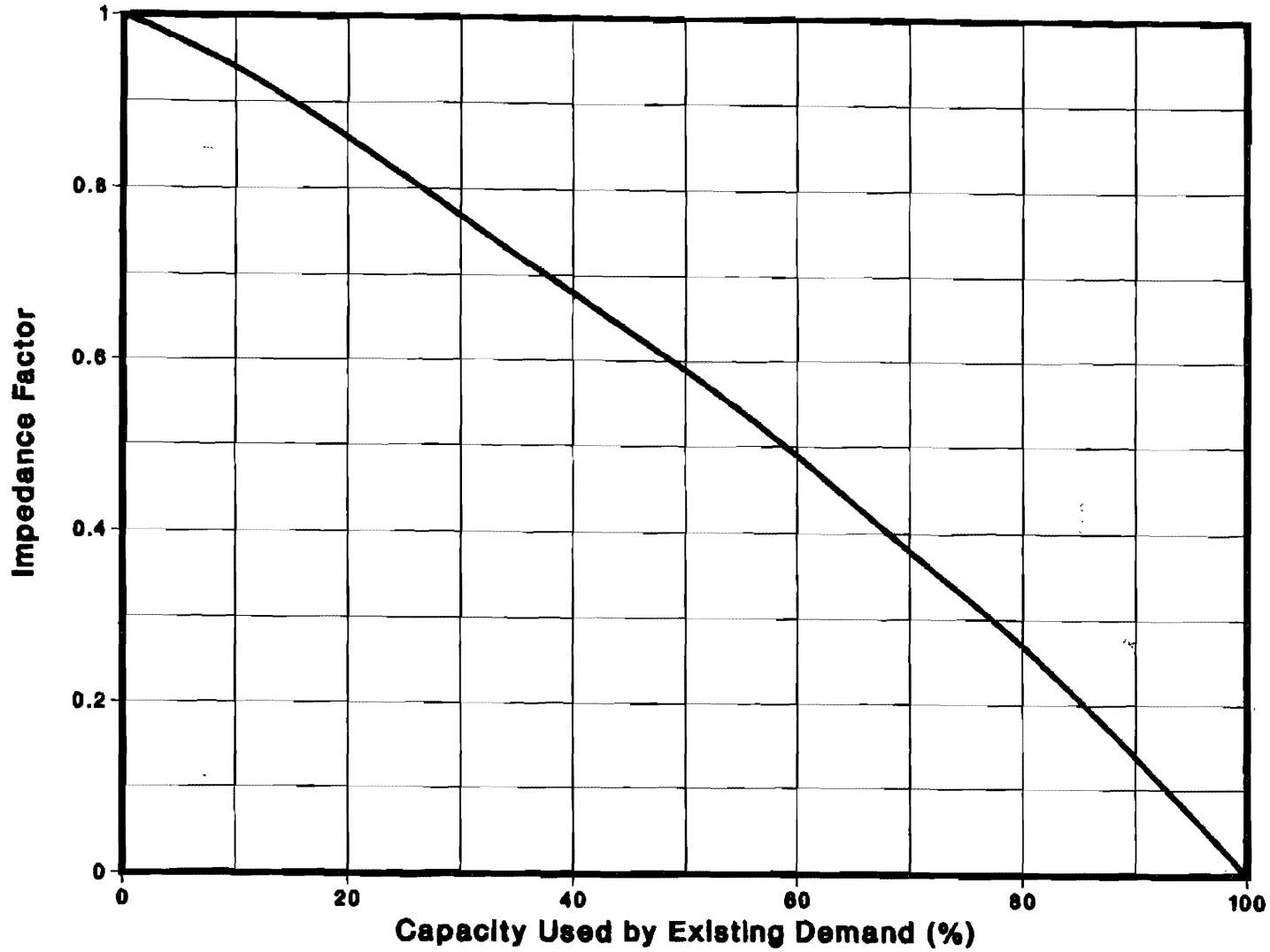
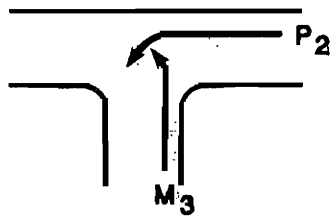


Figure 31

Capacity Reduction Caused by Congestion[49]

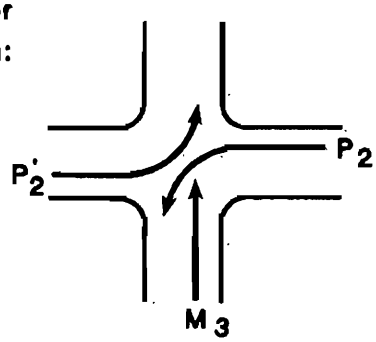
1. Left turns into the major street at a "T" intersection:

$$M_3 = M_{No} \times P_2$$



2. Thru traffic crossing the major street at a 4-way intersection:

$$M_3 = M_{No} \times P_2 \times P_2'$$



3. Left turns into the major street at a 4-way intersection:

$$M_4 = M_{No} \times P_2 \times P_2' \times P_1' \times P_3'$$

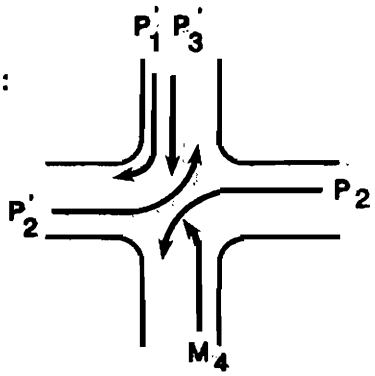


Figure 32

Application of Impedance Factors[49]

M_1 = capacity of the right streams

M_3 = capacity of the through streams

M_4 = capacity of the left streams

The difference between the calculated capacity and the existing demand, denoted as the reserve capacity, is directly proportional to the Level of Service as shown in Table 10. The reserve capacity for each roadway is taken as the weighted average of the reserve capacities for the individual movements on that roadway. As with signalized intersections, Table 7 relates Level of Service to stopped delay. Thus, a stopped delay per vehicle is determined for each leg of the intersection. Finally, for unsignalized intersections, queue lengths for individual legs of the intersection are calculated by:

$$QL = 8 \frac{\text{Existing demand}}{\text{Reserve Capacity}} \qquad (3-12)$$

For further details on the traffic algorithms used by TEXIN2, the reader is referred to *Estimates of Air Pollution Near Simple Signalized Intersections.*[5]

B. Determination of Vehicular Emissions

The second function performed by the TEXIN2 model is the estimation of vehicular emissions. The emissions are calculated as the sum of two components: cruise and excess emissions. Cruise and excess emissions are released by free-flowing and delayed vehicles, respectively. Initially, cruise emissions are assumed to be released along the entire length of each intersection leg. The emissions are subsequently redistributed to better reflect the traffic movement. A modified version of the MOBILE3[12] program is used to estimate cruise emissions and an idle emission factor, while excess emissions are calculated using procedures suggested by Ismart.[54] As an alternative, a short-cut method combining the MOBILE3 estimation of the idle emission factor with values for individual vehicle emission rates based on speed, temperature, the percent of hot/cold starts, and the vehicle scenario is available to the user.[55]

As modified for use in TEXIN2, MOBILE3 provides the user with inspection/maintenance (I/M) and anti-tampering program (ATP) options. Since the function of TEXIN2 is to estimate carbon monoxide concentrations near intersections, the vehicle emission factor calculational procedures for nitrogen oxides and hydrocarbons were deleted from MOBILE3. California data and options from MOBILE2 were added to MOBILE3 since these data are not initially available to the user. This was done in accord with suggestions of the MOBILE3 technical support staff in Ann Arbor, Michigan.[56] The MOBILE3 program includes the corrections to the model made by the EPA as described in the EPA memo of May 15, 1985.

The MOBILE3 program allows the user to apply I/M credits to the basic exhaust emission levels. The emission reduction credits attributable to an I/M program vary according to the

Table 10

Level of Service and Expected

Delay for Reserve Capacity Ranges[49]

Reserve Capacity	Level of Service	Expected Traffic Delay
400 or more	A	Little or no delay
300 to 399	B	Short traffic delay
200 to 299	C	Average traffic delay
100 to 199	D	Long traffic delay
0 to 99	E	Very long traffic delay
less than 0	E	Failure—Extreme congestion
(any value)	F	Intersection blocked by external causes

program type. The I/M credits depend on the following factors:

(1) Model years involved in the I/M program

(2) The calendar year being analyzed and the calendar year the I/M program was implemented

(3) The estimated first year failure rate or stringency level

(4) The vehicle types affected by the program

(5) The amount of mechanic training

(6) The type of I/M test implemented for 1981 and later light-duty vehicles

(7) The standards used in the I/M short test for 1981 and later light-duty vehicles

The additional inputs required for the I/M option are described in the Input/Output section of this chapter as well as the User's Guide.[57]

To compensate for the increase in tampering and its associated effect on fleet emission rates, MOBILE3 includes a correction term, which alters individual vehicle emission rates. Using this capability, the basic emission rates are calculated for untampered vehicles and the effects of tampering are included as offsets to those values. The offsets are determined from the percentage of vehicles being tampered with at any given time and the effects of such tampering. It is assumed that these effects grow linearly with mileage due to the observation that the frequency of tampering increases with vehicle age and accumulated mileage. The tampering effects are assumed to be independent of the mileage at which the vehicle was disabled.

The types of tampering included in the TEXIN2 model are:

(1) Misfueling (not applicable to fuel inlet disablement)

(2) Fuel inlet disablement

(3) Catalyst removal

(4) Air pump

Where applicable, any number of tampering types may apply to light-duty gasoline vehicles, light-duty gasoline trucks, and heavy-duty gasoline trucks. The default tampering frequencies are based on national averages, and differ for I/M and non-I/M areas. The user may also use local rates approved by the MOBILE3 technical support staff.[12]

ATP programs may also be taken into account using credits assigned to emission rates. The program allows for most types of ATP's, 102 of which are discussed in the EPA technical report EPA-AA-TSS-83-10.[58] The MOBILE3 User's Guide[12] includes credits for anti-tampering programs which inspect annually, biennially, upon change of ownership, or by random audits of 1%, 2%, and 5% of the vehicle fleet. Each option may include the inspection of a combination of one or more items, some of which EPA has determined credits for:

(1) Air pump only

(2) Air pump and catalyst

(3) Air pump, catalyst and fuel inlet

(4) Air pump, catalyst, and lead deposit test

(5) Catalyst only

(6) Catalyst and fuel inlet

(7) Catalyst, fuel inlet, and lead deposit test

When an ATP program is invoked, a different methodology is used in the calculation of emission rates. First the emission factors are calculated for the entire fleet without the effects of anti-tampering. Next, two separate calculations are made to evaluate the emission factors on the 1968 to 1979 and 1980 to 2020 fleets, respectively. These trials are done with the respective ATP coverage dates. These three values are then used to calculate the final emission factors applicable to a particular scenario.

The user may also correct the emission factors for various other factors. The impact of air conditioning usage may be determined from a knowledge of wet and dry bulb temperatures. The fraction of light duty vehicles towing trailers may be specified so that trailer towing corrections can be applied. Similarly, the user may specify the fraction of light duty vehicles with an extra 500 lb load so that extra loading corrections may be determined.

Alternatively, the user may utilize a *short-cut* method to determine the emission rates *in lieu* of the bulky MOBILE3 routine. The method was developed by combining portions of MOBILE3 with alternative cruise emission factors. The cruise emission factors are interpolated from the FHWA values[55] presented in Table 11, and adjusted for ambient temperature, as shown in Table 12. This adjustment is actually an incremental change in light-duty vehicle carbon monoxide emissions which is added to the initial base value. For each 10% increase in the number of non-catalyst equipped vehicles in the cold start mode and catalyst equipped vehicles in the hot start mode, a value of 0.2 grams/mile is added to the base value.

For catalyst equipped cold start vehicles, the incremental change is a function of ambient temperature and is interpolated between the values of 10.0 grams/mile at 20°F and 3.3 grams/mile at 75°F. For temperatures beyond the extremes, the limiting emission rates are used. The scheme for computing an idle emission factor was retained from MOBILE3.

The portion of the original model which calculates emissions due to vehicle slowing or stopping is retained in TEXIN2. A detailed discussion of this calculation is given by Messina, *et al.*[5]; however, the procedure is briefly discussed below. Ismart[54] gives the following equation for carbon monoxide emissions due to vehicles stopping:

$$COST = \frac{PCST \cdot TTEI \cdot ER}{1000} \qquad (3-13)$$

Table 11

1982 Carbon Monoxide Emissions at Various Speeds[55]

Data is in gm/mi†

Average Speed (mph)	LDGV	LDGT1	LDGT2	HDGV	LDDV LDDT HDDV
5	101.3	150.7	214.2	582.6	72.8
10	50.7	76.5	104.6	422.1	50.7
15	35.6	54.4	70.2	321.7	36.7
20	28.4	43.6	54.6	256.8	27.6
25	23.3	36.0	44.9	214.0	21.6
30	19.1	29.7	37.5	185.7	17.5
35	15.9	25.0	31.9	167.5	14.8
40	14.0	22.0	28.2	156.8	13.0
45	13.2	20.8	26.2	152.4	11.8
50	13.0	20.5	25.4	153.8	11.2
55	11.9	18.9	42.2	161.7	11.0

†100% hot stabilized

Table 12

Incremental Change in LDGV Carbon Monoxide Emissions for a 10 Percent Change in Hot/Cold Mode[55]

Mode	Incremental Change (gm/mi)
Catalyst Cold Mode	
20°F	10.0
75°F	3.3
Non-Catalyst Cold Mode	
20°F	0.2
75°F	0.2
Catalyst Hot Start	
20°F	0.2
75°F	0.2

where:

$COST$ = total amount of excess carbon monoxide emitted due to vehicles stopping (lb/hr)

ER = carbon monoxide emitted per 1000 speed changes (lb)

1000 = factor to convert ER to lbs per speed change

The emission rate is determined using Figure 33 by considering the vehicle as moving from an initial speed to zero and then returning to the initial speed. A correction factor is applied to account for the difference in scenarios used by Ismart and MOBILE3. This factor is calculated as the ratio of the MOBILE3 composite emission factor for the entered vehicle scenario to the MOBILE3 composite emission factor from the Modal Analysis Model vehicle scenario used by Ismart.

To determine the excess carbon monoxide emissions due to vehicle slowing, the following equation from Ismart[54] is applied to determine the time lost by vehicles slowing down but not stopping:

$$\text{Slowdown delay} = ADPV - TIQPV \qquad (3-14)$$

where:

$ADPV$ = approach delay (sec/veh)

$TIQPV$ = time in queue per vehicle (sec) given by:[53]

$$TIQPV = 1.282 \times SDPV - 0.628 \qquad (3-15)$$

Once the slowdown delay per vehicle is determined, the emissions due to slowing, $COSD$, are estimated from the equation by Ismart:[54]

$$COSD = \frac{(ADPV - TIQPV) \cdot TTEI \cdot ER}{3600 \cdot HRS} \qquad (3-16)$$

where:

HRS = excess hours consumed per 1000 speed changes (Table 13)

Again a correction factor is applied to the emission rate obtained from Figure 33. Ismart suggests that the slowdown speed is one-half the initial speed. Excess emissions due to vehicles idling, $COID$, are then calculated using the equation proposed by Ismart:

$$COID = (SDPV \cdot TTEI \cdot EFI/60) \cdot 453.6 \qquad (3-17)$$

where:

EFI = idle emission factor from MOBILE3 (gm/min)

$SDPV$ = stopped delay per vehicle (sec)

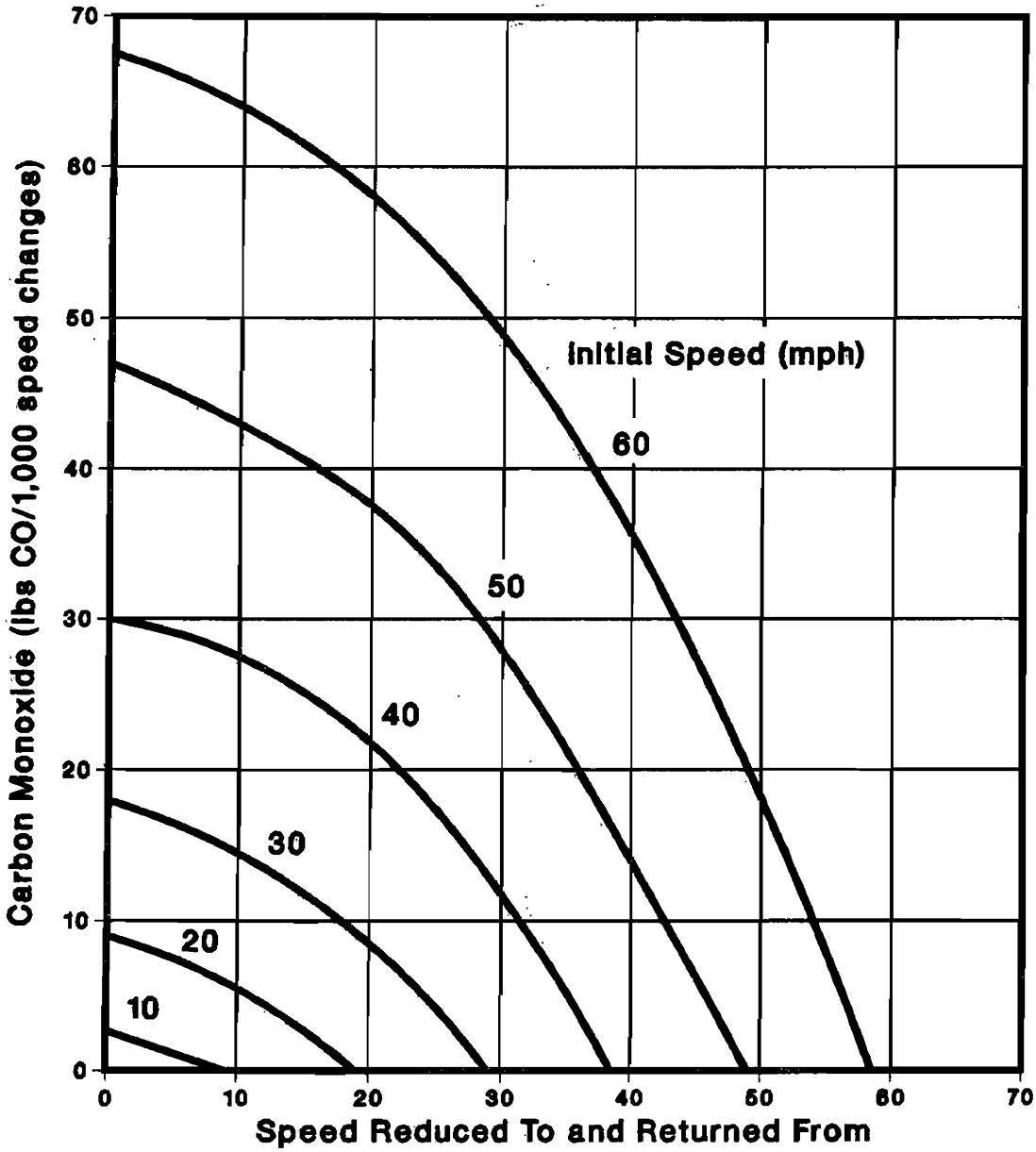


Figure 33

Carbon Monoxide Emissions for Vehicular Speed Changes[54]

Table 13

Excess Hours Consumed for Vehicular Speed Changes[59]

hrs/1000 speed changes

Initial Speed	Speed Reduced to and Returned From											
(mph)	Stop	5 8	10 16	15 24	20 32	25 40	30 48	35 56	40 64	45 72	50 80	(mph) (kph)
5	1.02											
10	1.51	0.62										
15	2.00	1.12	0.46									
20	2.49	1.62	0.93	0.35								
25	2.98	2.11	1.40	0.80	0.28							
30	3.46	2.60	1.87	1.24	0.70	0.23						
35	3.94	3.09	2.34	1.69	1.11	0.60	0.19					
40	4.42	3.58	2.81	2.13	1.52	0.97	0.51	0.16				
45	4.90	4.06	3.28	2.57	1.93	1.34	0.83	0.42	0.13			
50	5.37	4.54	3.75	3.01	2.34	1.71	1.15	0.68	0.35	0.11		
55	5.84	5.02	4.21	3.45	2.74	2.08	1.47	0.94	0.57	0.28	0.09	

Finally, the total excess emission factor is calculated from the values for $COST$, $COSD$, $COID$, and the total queue length QL by:

$$EF = \frac{(COST + COID + COSD) \cdot 453.6}{QL \cdot 3600} \qquad (3-18)$$

where the emission factor, EF is in gm/m · sec.

Since the CMA treats the entire intersection as a whole, the model does not distinguish between the various approach legs when determining excess emissions. One excess emission factor, applicable to all legs, is calculated for the entire intersection. However, the excess emissions distribution routine treats each leg individually. The specific queue length associated with each leg is used in the length of roadway over which the excess emissions are emitted for that leg. For an unsignalized intersection, specific excess emission factors are applied to each leg in accordance with the CMA technique for such intersection scenarios.

In TEXIN2, once the emissions have been assigned to the appropriate links and pseudolinks as described above, a redistribution of emissions is enacted. Cruise emissions are treated separately from idle emissions and excess emissions due to acceleration and deceleration. Furthermore, each type of intersection, signalized or unsignalized, is handled separately.

Idle emissions are assigned to the appropriate pseudolink and, since no traffic movement is involved, redistribution is not necessary. As modeled in TEXIN, excess emissions due to slowing, stopping, and accelerating are assigned to the pseudolink consistent with the approach link. In TEXIN2, only emissions due to slowing and stopping are applied to this pseudolink. Once the vehicle accelerates, the emissions are spread to the pseudolink upon which the vehicle exits the intersection. This keeps the distribution of those excess emissions consistent with traffic flow.

Cruise emissions are also redistributed according to traffic flow. In addition to the emissions from inbound and outbound traffic on each leg, emissions due to turning vehicles are also included in TEXIN2. Turning vehicles are assumed to cruise through the turn at 10 mph from an initial spot in the queue or to slow down to 10 mph while approaching the turn. The proportions used are equivalent to the fraction stopping and one less the fraction stopping, respectively. These emissions are then equally distributed along the approach and exit legs of the turning vehicle. Emissions from links further away from the intersection are not modified, but remain a result of inbound and outbound traffic on the link.

C. Modeling Atmospheric Dispersion

TEXIN2 utilizes the same dispersion model as the original model, CALINE3.[60] This dispersion model is a Gaussian dispersion model and is quite similar to the dispersion routines used by CALINE4. Therefore, the reader is referred to the CALINE4 section in Chapter 2 for a brief description of the model.

One modification made to CALINE3 is presented below. Since CALINE3 does not adequately simulate dispersion under low wind speeds, a correction factor must be applied in such cases. Since Chock[61] reports an ambient plume rise of 0.15 m/sec for a 0 m/sec crosswind, Nelli[5] originally proposed the following equation for wind speeds below 1 m/sec:

$$H = 0.15 \cdot TRES \qquad (3-19)$$

H can be thought of as the height that a pollutant emitted at the roadway centerline would rise by the time it reached the roadway edge. $TRES$ is the pollutant residence time calculated by CALINE3.

The model incorporates a worst case wind angle analysis. By appropriately assigning the flag $WCFLAG$, the model will perform a search to determine at what wind angle the highest carbon monoxide concentrations at each receptor are observed.

D. Modeling Miscellaneous Intersections

Aside from the basic four-leg signalized and unsignalized intersection, three other types of intersections can be modeled by TEXIN2: T-intersections, one-way streets, and 4×4 four-way stops. Minor changes were needed to model T-intersections while one-way streets are modeled merely by assigning no inbound traffic entering the intersection on the leg which traffic leaves the intersection. The speed used for this leg of the intersection is the speed at which traffic leaves the intersection on that leg. The addition of the flag $TFLAG$, enabled the modeling of the dispersion process for T-intersections. The value of the *missing* leg (1 through 4) is assigned to this flag while modeling such intersections. A zero value of $TFLAG$ signifies a normal four-leg intersection. When simulating T-intersections, the missing leg is internally assigned default values to circumvent calculation errors such as division by zero. As needed, the emission factor associated with the absent link (and the associated pseudolink) is set to zero. The CMA Operations and Design procedure is recommended (but not mandatory) for signalized T-intersections.

The algorithm for four-way intersections presented by Lee, *et al.*[13], is used to describe both the traffic flow and emissions with this type of intersection. Three variables are needed to model the intersection: volume of approaching traffic, number of left turns, and truck percentages. Each variable is assigned a value of -1, 0, or $+1$ for low, intermediate, and high levels, respectively. The specific ranges for the three levels, as shown in Table 14, were found using the assumption that an all-way stop sign controlled intersection cannot, in general, process more than 2500 vehicles per hour. Inherent to this assumption is the absence of left-turning vehicles and trucks.

The traffic entering and leaving via each leg is initially established, and the traffic flow analysis and emission rates are subsequently determined. One should be reminded that the traffic flow analysis is not necessary for the calculation of emissions. Tables 15 and 16 describe the parameters

Table 14

Specific Ranges for the Three Levels of the Four Way Stop Parameters[13]

Parameter	Range Low	Intermediate	High
Approach Volume (vehicles)	< 250	250 to 500	> 500
Left Turns (vehicles)	< 48	48 to 96	> 96
Trucks (percent)	< 5	5 to 10	> 10

and equations used in the analysis of traffic parameters and emissions. The equations are based on experimental runs of 250 and 500 vehicles per hour per lane (vphpl) on 12 foot lanes with 100 right turns per hour and bucket lengths of 100 ft.

Each intersection leg is modeled as the sum of emissions on the approach lanes and the outbound lanes. Each lane is divided into 100 foot segments and the segments summed for each link. Since there is an increased amount of time spent in the immediate intersection area, the segment nearest the intersection is treated individually for each approach leg of the intersection.

E. Summary of Data Required in TEXIN2

The input requirements for TEXIN2 can be divided into four general categories: link description, receptor coordinates, meteorological conditions, and vehicle scenario. If the I/M and/or ATP options of MOBILE3 are invoked, additional parameters will be needed to describe those programs. To assign receptor coordinates to the required locations, the center of the intersection should be placed at the origin of an $x - y$ Cartesian system. The northernmost leg of the intersection is then aligned with the $y-$axis.

The first input information in the model concerns the physical descriptions of the individual legs of the major intersection as well as minor side streets. Since the model treats each leg as a link, individual lanes need not be addressed. Parameters required to fully describe each link include:

(1) Coordinates in the $x - y$ system
(2) Width
(3) Link type, (*i.e.*, at grade, fill bridge, *etc.*)
(4) Traffic volume
(5) Average speed on non-delayed vehicles
(6) Number of approach and turn lanes
(7) Estimated percentage of turning vehicles (left and right)
(8) Source (link) height
(9) Width of through and left turn lanes

Certain aspects of the intersection operation must also be stated including the number of signal phases, left-turn phases, and cycle length.

The remaining input parameters concern the receptors, meteorology, and vehicle scenario. Three dimensional Cartesian coordinates must be specified for each receptor. Meteorological data include wind speed, wind direction, stability class, temperature, mixing height, surface roughness, ambient carbon monoxide concentration, and averaging time. The percentage of hot/cold starts must also be specified to estimate emissions.

Finally, additional data must be supplied to satisfy the required options. For example, the VMT (vehicle miles traveled) mix may be specified in place of the national default values. Addi-

Table 15

Description of Parameters Used in the Four Way Stop Algorithm[13]

Symbol	Approach Responses	Units
ATDA	Average total delay of all vehicles on inbound approach	sec/veh
ATDL	Average total delay of left turns on inbound approach	sec/veh
ATDR	Average total delay of right turns on inbound approach	sec/veh
ATDS	Average total delay of straights on inbound approach	sec/veh
ASDA	Average stop delay of all vehicles on inbound approach	sec/veh
ASDL	Average stop delay of left turns on inbound approach	sec/veh
ASDR	Average stop delay of right turns on inbound approach	sec/veh
ASDS	Average stop delay of straights on inbound approach	sec/veh
QAVG	Average queue length on approach	Number of vehicles
QMAX	Maximum queue length on approach	Number of vehicles
COOP	Total CO emission on outbound approach	kg/15 min
COIP	Total CO emission on inbound approach	kg/15 min
COBK	Total CO emission on bucket nearest intersection	kg/15 min

NOTE: All buckets are 100 ft. long

Table 16

Equations Used in the Four Way Stop Algorithm[13]

Response	Predictive Model
ATDA	$16.67 + 4.33 \times VO + 1.37 \times LT + 1.15 \times VO \times LT$
ATDL	$9.14 + 2.78 \times VO + 9.14 \times LT + 2.78 \times VO \times LT - .73 \times VO \times TR - .73 \times VO \times LT \times TR$
ATDR	$17.29 + 4.96 \times VO + 1.51 \times LT + 1.55 \times VO \times LT$
ATDS	$16.44 + 4.11 \times VO + 1.29 \times LT + .94 \times VO \times LT$
ASDA	$8.05 + 1.65 \times VO + .80 \times LT + .68 \times VO \times LT$
ASDL	$4.58 + 1.33 \times VO + 4.58 \times LT - .32 \times TR + 1.33 \times VO \times LT - .31 \times VO \times TR$ $-.32 \times LT \times TR - .31 \times VO \times LT \times TR$
ASDR	$7.82 + 1.73 \times VO + .75 \times LT - .41 \times TR + .69 \times VO \times LT - .44 \times VO \times TR$ $-.43 \times LT \times TR - .37 \times VO \times LT \times TR$
ASDS	$8.14 + 1.50 \times VO + .81 \times LT + .59 \times VO \times LT$
QAVG	$1.14 + .57 \times VO + .15 \times LT + .14 \times VO \times LT$
QMAX	$2.91 + .81 \times VO + .34 \times LT$
COOP	$258.62 + 135.75 \times TR$
COIP	$415.24 + 149.92 \times VO + 97.32 \times TR$
COBK	$102.47 + 42.17 \times VO + 6.04 \times LT + 5.88 \times VO \times LT - 3.70 \times VO \times LT \times TR$
COOP	$258.62 + 135.75 \times TR$

where:

$VO =$ Volume of traffic
$TR =$ Truck volume
$LT =$ Fraction of left turning vehicles

tionally, the user may specify local values of the mileage/registration distribution and optional air conditioning usage data, trailer towing fractions, and extra loading fractions. If the I/M program is utilized, additional input includes:

(1) Year of I/M program implementation

(2) Stringency level of the I/M program

(3) Mechanic training as a part of program effectiveness

(4) Earliest and latest models year included in the program

(5) Type of vehicles covered by I/M

(6) Type of I/M test (and its standards) implemented for 1981 and later light-duty gas vehicles

The anti-tampering option required the following additional data:

(1) Year of ATP implementation

(2) First and last model years covered by the ATP

(3) Vehicle classes included in the ATP

(4) Type of ATP and its associated credit rates

(5) Tampering rates

If the latter item is not specified, national default values are used.

The primary output of concern from the TEXIN2 model is the calculated carbon monoxide concentrations at each receptor. Additionally, carbon monoxide concentrations contributed by each link and pseudolink, summary of input data, composite emission factors and idle emission rates, excess emission factors and traffic data are available at the request of the user. For a detailed description on the implementation of TEXIN2, consult the User's Guide.[57]

Chapter 4

Site Description

The research site was located on the northwest side of Houston on loop IH610 between Airline Drive and North Main Street, approximately one mile west of the IH45 and IH610 interchange. Selection of this site was made on the basis of several criteria including right of way width, access to storm drainage systems for placement of electrical cable, and availability of sign bridges for the location of radar units. An overhead view of the research site is shown in Figure 34 and the equipment layout used in project 283 is shown in Figure 35.

The site chosen for this project was also utilized by TTI in an similar study[11] in 1976. The use of this site would allow a comparison to be made between pollutant concentrations in 1976 and 1984. This would provide the basis for a valuable data base for use in roadway air quality model verification and development. Furthermore, it was desired to monitor ozone, hydrocarbons, and oxides of nitrogen in addition to carbon monoxide while the 1976 study concentrated primarily on carbon monoxide.

The freeway ran from the west-southwest to the east-northeast at a compass heading of 78°. Since the prevailing winds in the area were from the south, the majority of the instrumentation could be located on the north side of the roadway where the right of way was the widest. Traffic on IH610 was usually moderate being the heaviest around 7:30–8:15 a.m. and 3:15–5:30 p.m. Westbound traffic rarely was severely impeded but eastbound traffic was slow in the afternoon especially in the inside lanes during several monitoring periods. This was attributed to construction on IH45 north into which the inside lanes merged at the nearby interchange. The freeway consisted of four through traffic lanes and one exit ramp lane in both directions. The lane width was 12 feet and the center barrier width was 20 feet. Two lane frontage roads paralleled the freeway on both sides. The north frontage road was lightly traveled but traffic on the south frontage road would occasionally become heavy when the traffic on IH45 was congested.

The mobile environmental laboratories (MEL) used in the project were parked on both sides of the freeway where Link Road intersected the frontage roads. On the north side of the road, two laboratories were used (MEL-2 and MEL-3). On the south side, the laboratory AEDAS-1 (Automated Environmental Data Acquisition System) was used. The laboratories were fenced off on both sides of the freeway and meteorological towers were located within the fenced area. The tower on the south side of IH610 had receptor heights of 55, 37, and 19 feet. The tower on the north side had receptor heights of 102, 74, 47, 33, and 5 feet. Electrical cable placed in drainage systems below the freeway allowed data from UVW anemometers to be collected by a computer located in MEL-2 since the software on the computer in the AEDAS was not equipped to monitor those instruments.

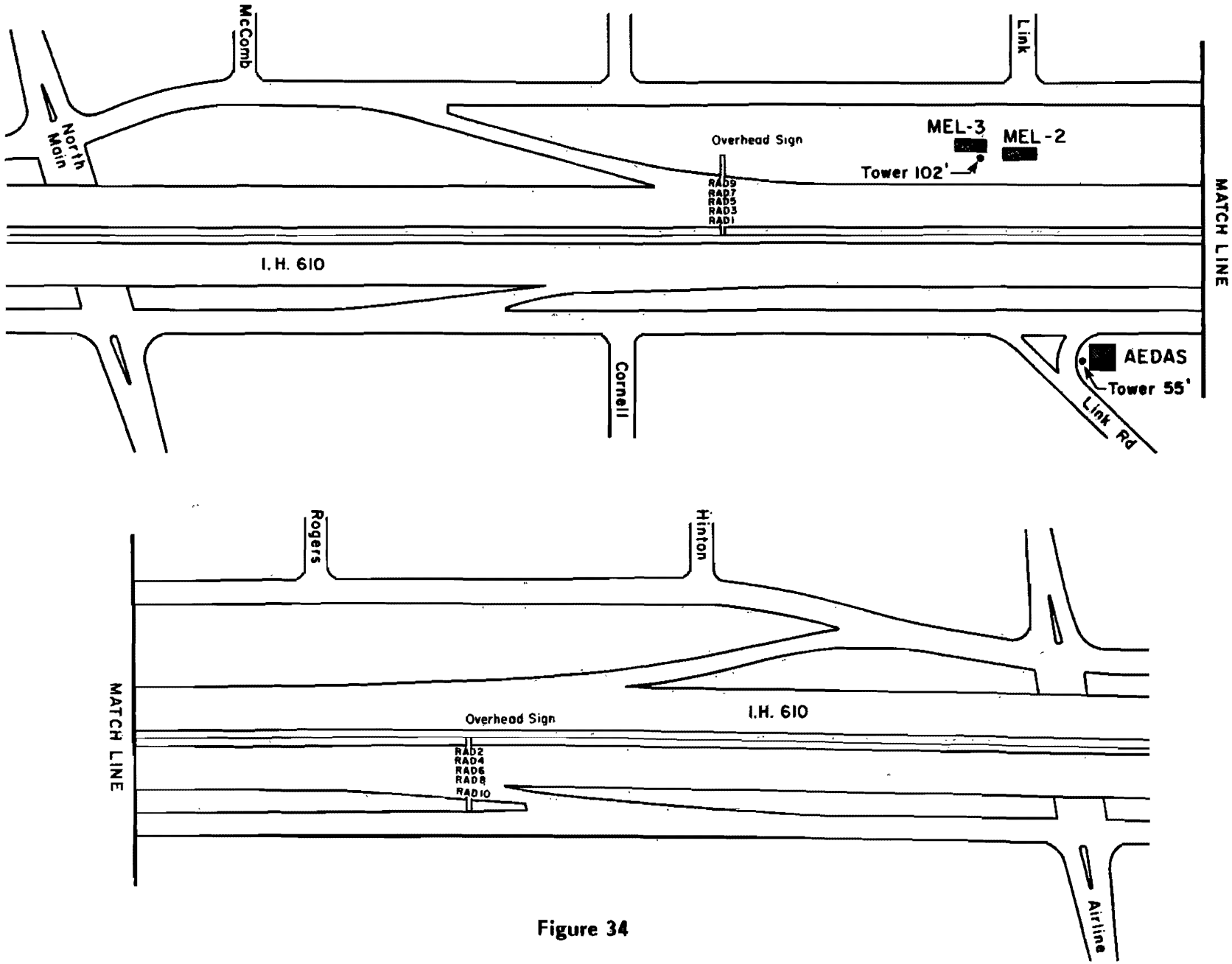


Figure 34

Overhead View of the Houston Research Site—IH610 at Link Road

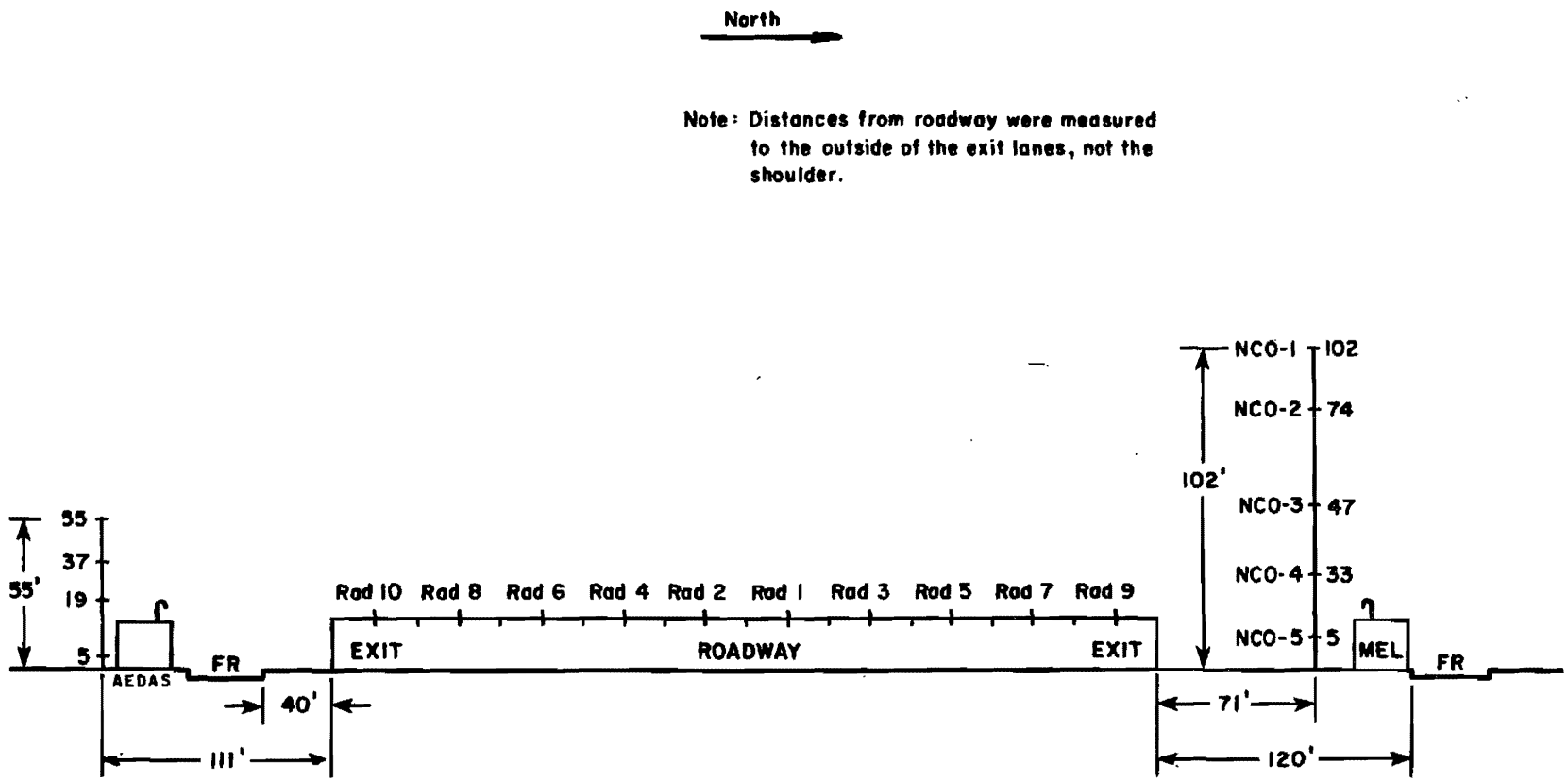


Figure 35

Cross Section of the Houston Research Site—IH610 at Link Road

The surrounding area consisted of single story dwellings. There were trees up to 60 feet tall on both sides of the roadway. There was no industry located in the immediate area. The right of ways were relatively smooth with lightly scattered trees up to 20 feet tall. The research site was an at-grade site.

Chapter 5

Experimental Methods

Over 140 hours of data were collected at the Houston research site. Although the primary emphasis was on traffic, meteorological and carbon monoxide data, other pollutants including hydrocarbons, nitrogen oxides, and ozone were also monitored. This section describes the instruments used to collect the data.

I. Data Acquisition System

Two computers, one in MEL-2 and one in AEDAS-1, were used to monitor all instruments. The computer in MEL-2 was assembled by Balcones Computer Corporation. It consisted of a Z-80 microprocessor, two 32-channel, 12 bit analog-to-digital converters, a real time clock, and a nine-track magnetic tape drive. The operating system was CP/M. The majority of the instruments were interfaced to this computer. The software in this computer was written locally so that changes were not extremely difficult to perform. Table 17 lists the instrumentation on this computer and the respective sample rates.

The second computer was a Radian DART. This machine was located in AEDAS-1. All instruments on the south side of the roadway were interfaced to the DART, except the UVW anemometers. This computer lacked the flexibilities of the Balcones computer and, because of inadequate documentation on the software, many available features were never utilized. Table 18 summarizes the instruments interfaced to the DART and the respective sample rates.

II. Traffic Measurement

A large effort was made to accurately measure traffic flow in Project 283. The method utilized Stephenson Mark V doppler shift radars originally obtained from the Texas Department of Public Safety. The experimental procedure was originally implemented by Bullin and Polasek[62] and modified slightly due to different data acquisition systems and electrical problems with the radar signals.

The radar units consisted of a control cabinet and an external antenna. The antennas were modified so that they could be mounted on the sign bridge and directed at traffic at a 45° angle. Sheet metal shelters were constructed to keep the electronics control unit out of the weather. Five radars were mounted on each bridge, over each lane so that the antennas faced directly towards oncoming traffic. A variable DC power supply, capable of delivering at least 120 watts, was used for radar power sources on each bridge.

The radar units had a speed indicator meter and an analog voltage output (0–10 VDC) that was proportional to the component of the speed moving directly towards the face of the antenna.

Table 17

Instrumentation Used on the Balcones Computer

Channel	Instrument	Sample Rate (sec)	Sensor Height (ft)	Record Type
1	Rad 1	0.01	—	10
2	Rad 2	0.01	—	10
3	Rad 3	0.01	—	10
4	Rad 4	0.01	—	10
5	Rad 5	0.01	—	10
6	Rad 6	0.01	—	10
7	Rad 7	0.01	—	10
8	Rad 8	0.01	—	10
9	Rad 9	0.01	—	10
10	Rad 10	0.01	—	10
11	N UVW Perpendicular	10.0	47	19
12	DASIBI CO	2.0	12	7
13	N UVW Parallel	10.0	47	19
14	DASIBI O_3	20.0	12	3
15	N UVW Vertical	5.0	47	19
16	TECO NO_x	20.0	12	4
17	N UVW Perpendicular	10.0	33	19
19	N UVW Parallel	10.0	33	19
20	TECO NO_2	20.0	12	4
21	N UVW Vertical	5.0	33	19
22	TECO NO	20.0	12	4
23	Horizontal Anemometer	10.0	102	12
24	Byron THC	60.0	12	5
25	Wind Vane	5.0	102	13
26	Byron NMHC	60.0	12	5
27	Temperature	60.0	5	14
28	Byron CH_4	60.0	12	5
29	Wind Vane	5.0	74	13
30	Byron CO	60.0	12	5
31	Temperature	60.0	74	14
32	Byron CO_2	60.0	12	5
35	Horizontal Anemometer	10.0	74	12
37	N Vertical Anem.	5.0	102	18

Table 17 (Continued)

Channel	Instrument	Sample Rate (sec)	Sensor Height (ft)	Record Type
38	S UVW Perpendicular	10.0	19	19
39	N Vertical Anem.	4.0	74	18
40	S UVW Parallel	10.0	19	19
41	N UVW Perpendicular	10.0	5	19
42	S UVW Vertical	5.0	19	19
43	N UVW Parallel	10.0	5	19
45	N UVW Vertical	5.0	5	19
49	Ecolyzer	2.0	102	17
50	S UVW Perpendicular	10.0	37	19
51	Ecolyzer	2.0	74	17
52	S UVW Parallel	10.0	37	19
53	Ecolyzer	2.0	47	17
54	S UVW Vertical	5.0	37	19
55	Ecolyzer	2.0	33	17
56	S UVW Perpendicular	10.0	5	19
57	Ecolyzer	2.0	5	17
58	S UVW Parallel	10.0	5	19
59	Humidity	60.0	25	15
60	S UVW Vertical	5.0	5	19
61	Insolation	60.0	12	16
64	S Vertical Anem.	5.0	55	18

NOTE: All UVW anemometers listed above include the tower in which the instrument was located (N—North; S—South).

Table 18

Instrumentation Used on the DART Computer

Octal Channel	Instrument	Sensor Height (ft)
00	DASIBI CO	12
01	DASIBI O_3	12
02	TECO NO_x	12
03	TECO NO	12
04†	Byron THC	12
11	TECO NO_2	12
12	Insolation	12
15	Anemometer	55
17	Wind Vane	55
22	Temperature	5
23	Temperature	37
24	Temperature	Interior
25	Barometer	12
31	Ecolyzer	55

†Indicates that a limited amount of data is available for this instrument.

NOTE: All instruments were sample at 1 second rates by the DART.

This analog signal was sampled every 0.01 seconds by the Balcones computer. The A/D on the computer was hardware selected for a voltage range of -5 to $+5$ VDC full scale; therefore, the radar output signal had to be attenuated by a factor of two. Since traffic moving at normal speeds dwell only a fraction of a second under the radar beam, the indicator needle on the radar did not respond fast enough to adequately measure the vehicular speed. However, due to the fast sample rate of the radars, the computer was able to sense the entire response of the analog output.

The radar sent essentially a square voltage pulse to the computer for each vehicle passing under the radar. The magnitude of the voltage signal was proportional to the speed of the vehicle and the number of pulses equal to the number of vehicles which resulted in a highly accurate traffic monitoring system. Since the pulse was essentially square, it could be integrated to obtain the length of the vehicle. This allowed the traffic to be divided into five categories based purely on length. These categories were selected by project personnel so that they involved distinct types of vehicles:

(1) Small cars (Category 1)
(2) Large cars, vans, and light duty trucks (Category 2)
(3) Medium duty trucks (Category 3)
(4) Tractor trailers and other heavy duty vehicles (Category 4)
(5) Tailgaters and radar calibration data (Category 5)

A primitive method was used to integrate the radar signal which essentially involved a summation of the voltage signal samples while the vehicle was under the radar. Once the vehicle left the radar, the resulting sum was compared to a table of four numbers. Since the summation was a function of length only, the comparison could be used for classification. This method of integration was used because at the high sample rate of the radars, the computer did not have time to use more elaborate methods. Furthermore, since the signal was square, this technique proved to be quite accurate. The speed of the vehicle was taken as the largest single sample while the vehicle was under the radar. Since the radars only reported the component of the speed moving directly towards the unit, the calculated speed and area had to be corrected by a factor of the square root of two. At the end of each minute, the traffic data were averaged and placed in a magnetic tape buffer in the memory of the computer.

Radar calibration was done frequently since the process was fairly easy. Each radar had an internal tuning fork that was activated during calibration. This tuning fork resonated at a frequency that was equal to the doppler shift frequency of a 65 mph vehicle. Therefore, after activating the calibration mode, the output speed of the radar was adjusted to 65 mph by a potentiometer. The range control on the radar was adjusted as needed by project personnel. This control was much more difficult to set than the calibration. This control determined the quantity of the reflected

radar beam that would be processed by the electronics unit. The range setting was critical to the integration of the radar output since it directly affected the effective beam pattern on the roadway. An attempt was made to use the smallest apparent range setting possible for an average radar since this range would produce the best separation between close vehicles and smallest elliptical pattern on the road. However, the setting could not be too small or the radar would fail to process the returned microwave signal. The exact range setting was set by a digital integration program on the Balcones computer. However, since this program could only be executed outside of monitoring periods, other range setting procedures were employed at times. These methods involved the use of an analog integrator and visual inspection of the response of the radar indicator meter. After enough experience, visual inspection could often be used to adequately set the range. However, the most accurate method was the digital integration by the computer since the same algorithm was employed in the actual data acquisition software. Therefore, the range was usually adjusted after the monitoring period was over.

III. Meteorological Measurements

A. Wind Speed and Direction

Depending upon the monitoring station, two types of wind speed and direction instruments were used. Near the top of the towers, six cup Texas Electronics anemometers and wind vanes were used. They were located at heights of 102 and 74 feet on the north tower and 55 feet on the south tower. These anemometers used a photo chopper to measure the wind speed. A 60 slot disc rotated concurrently with the cups and interrupted an infrared beam emitted by a light emitting diode (L.E.D.). The optimum starting threshold of these instruments was 0.75 mph and they had a full scale accuracy of $\pm 2\%$. The Texas Electronic wind vanes used a constant rotation potentiometer to regulate an output signal that was proportional to the azimuth wind angle. The wind vanes had a 0–360° full scale corresponding to an output of 0–1.0 VDC. These vanes had a 3° dead band and less than a 2.0 mph starting threshold at 10° vane release angle. The linearity was $\pm 0.5\%$.

At heights of 47, 33, and 5 feet on the north tower and 37, 19, and 5 feet on the south tower, the wind speed and direction were measured with Gill UVW anemometers from RM Young Company and Weather Measure. These instruments allowed for simultaneous measurement of the three wind components. The UVW anemometers produced a signal which was directly proportional to the wind speed of the respective component. This meant that signal polarity was an important factor in determining wind direction. Since the three components were measured simultaneously, a vector sum on the components allowed for the calculation of the azimuth and elevation angles in addition to the wind speed. The starting threshold of these instruments depended upon the material used to construct the propellers. An attempt was made to use all polystyrene propellers, but due to a

broken mold, the manufacturer was not able to supply enough of these propellers. Therefore, on the lowest levels on each side of IH610 polystyrene propellers were used because they had the smallest starting threshold and the wind speeds near ground level were expected to be the lowest. The starting threshold of the polystyrene propellers was 0.3–0.5 mph. At other heights, polypropylene propellers were used. These propellers had starting thresholds that were about twice that of the polystyrene propellers.

At locations where cup anemometers were used to measure the horizontal wind speed, single propeller anemometers were added so that the vertical speed could also be determined. These instruments were a single component of the UVW anemometer and hence had the characteristics of the UVW anemometer.

B. Atmospheric Temperature and Humidity

To obtain information on atmospheric stability, temperature measurements were made with Texas Electronics Model No. 2015 thermistors. These instruments were located at heights of 74 and 5 feet on the north side and 33 and 5 feet on the south side. The manufacturer claimed these instruments had an accuracy of $\pm 0.5\%$ of full scale.

Relative humidity measurements were made on both sides of the roadway using Texas Electronics Model TH-2013 psychrometer. An increase in relative humidity would increase the length of hair sensor element which would in turn change the position of the suspended core in a linear variable differential transformer. Therefore, the output AC voltage of the transformer would change with changes in relative humidity. This AC voltage was then conditioned and rectified to produce a DC voltage proportional to the relative humidity. The accuracy of this instrument was $\pm 3\%$ relative humidity.

C. Solar Radiation

Solar radiation was measured on both sides of the roadway using Texas Electronics Model TS-100 photo voltaic pyranometers. The pyranometer produced an output voltage that was amplified by a Texas Electronics amplifier. The accuracy of the sensor was $\pm 10\%$ of the industry standard 48 junction thermopile black and white pyranometer with the error occurring at low solar incidence. The sensor was mounted in a glass dome to obtain the maximum cosine response of the angle of incidence. This instrument proved to be quite trouble free.

D. Barometric Pressure

Barometric pressure was measured on the south side of IH610 using a Texas Electronics Model No. TB-1012 Barometric Pressure Sensor. The sensor consisted of a bellows system which was directly coupled to the core of a linear variable differential transformer as in the psychrometers. The accuracy of the instrument was ± 0.02 inches of mercury over a ± 2.00 in. Hg span.

IV. Pollutant Concentration Measurements

A. Ozone Monitoring

Ozone samples were withdrawn from glass manifolds in MEL-2 and AEDAS-1. Figure 36 illustrates the sampling system used in MEL-2 and AEDAS-1. These manifolds were attached to a fan that pulled air into the manifold from a sample cane in the roof of each laboratory. The effective sample height of the cane was about 15 feet. The ozone samples were the first samples drawn from the manifold in order to minimize the effects of the rapid decomposition of ozone. The air samples were analyzed using two DASIBI Environmental Model 1003 ozone analyzers. The analyzers compared the amount of 253.7 nm ultraviolet light absorption in an air sample to that of an air sample that was scrubbed of ozone. These instruments had an accuracy of ± 0.001 ppm. The output signal from the analyzer in MEL-2 was amplified by 50 so that it could be intelligible to the A/D in the Balcones. Frequent checks on the amplifier proved that it was quite reproducible and linear.

B. Nitrogen Oxides Monitoring

Two Thermo Electron Model 14D monitors (EPA designated reference method) were used to measure the concentration of nitrogen oxides (NO_x, NO_2, and NO) in the ambient air. The analyzers sampled air from the same manifold that the ozone was taken. The instruments employed the photometric detection of the chemiluminescent reaction between nitric oxide (NO) and ozone:

$$O_3 + NO \longrightarrow NO_2 + O_2 + h\nu \qquad (5-1)$$

The instruments contained an ultraviolet light source that produced excess ozone from oxygen in dried ambient air. This ozone was fed to the reaction chambers where it reacted with the NO. As samples entered the instrument, they were split into two streams, one leading directly to the reaction chamber for the direct measurement of the NO concentration, and the other leading to a molybdenum catalyst which reduced all NO_x to NO. The output gases from the catalytic converter were introduced into another reaction chamber where the chemiluminescent reaction took place. In this chamber NO_x was measured so that NO_2 concentrations could be calculated by electronic subtraction of NO_x and NO signals. The accuracy of these analyzers was $\pm 2\%$ with a zero drift of ± 0.002 ppm. The NO_x analyzers were calibrated using a dynamic calibrator that mixed NO_x free zero air with a span gas containing NO in nitrogen. These calibrators also contained an ultraviolet lamp so that oxygen could be ozonated and react with part of the diluted NO. Therefore, known concentrations of NO_x, NO_2, and NO were introduced into the analyzer during calibration.

C. Hydrocarbon Sensors

Two Byron Model 401 gas chromatographs were used to monitor hydrocarbons. The entire analytical cycle on these instruments would monitor total hydrocarbons, non-methane hydrocarbons, carbon monoxide, methane, and carbon dioxide. These samplers were located in MEL-2 and

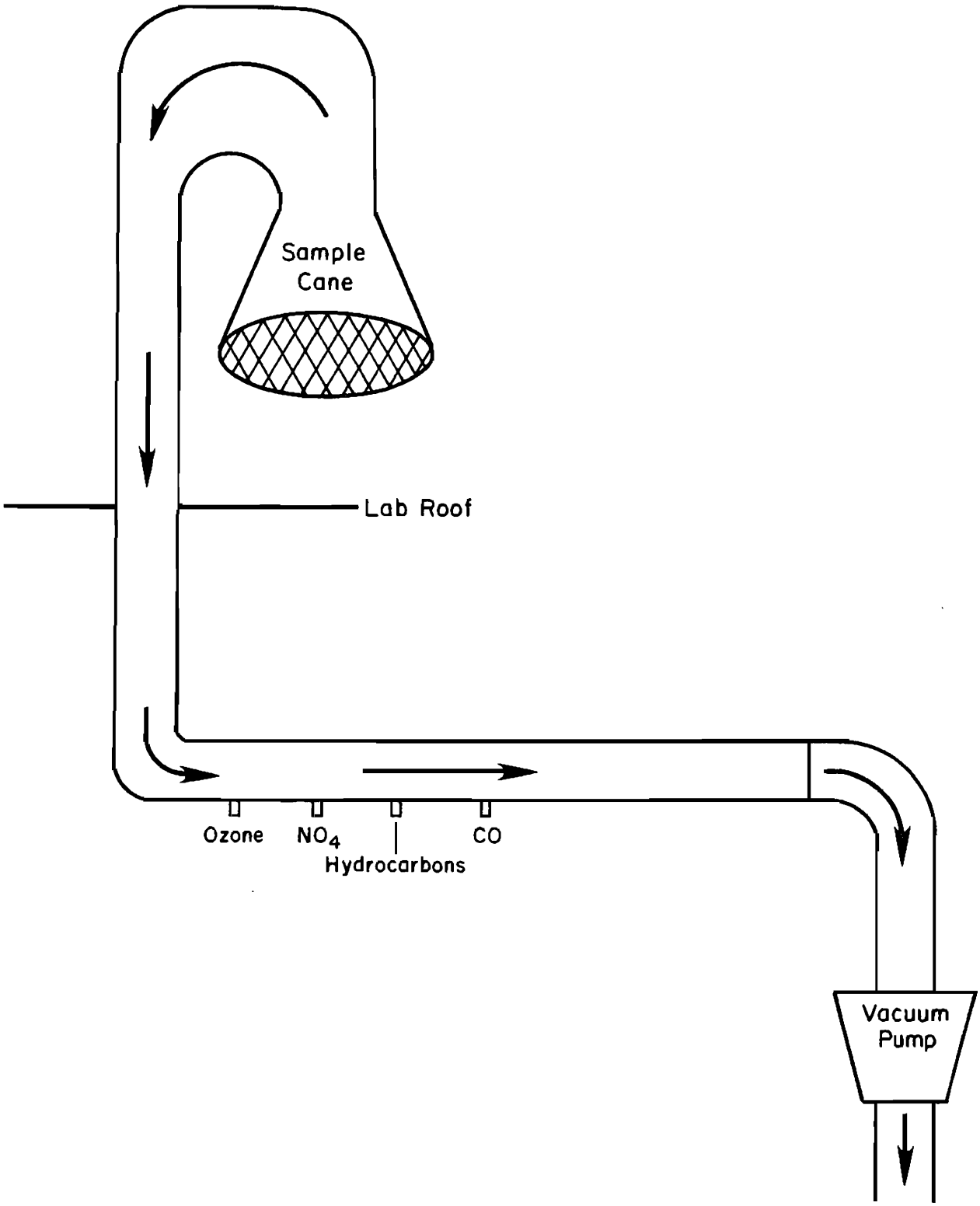


Figure 36

Sampling System Utilized in the Mobile Laboratories

AEDAS-1. The carrier gas for these chromatographs was generated by a Byron Model 25 Pure Air System. The carrier had to have less than 0.1 ppm of THC, CO, and CO_2. The chromatographs employed flame ionization detectors (FID) and used hydrogen as the fuel. The hydrogen was supplied by a General Electric hydrogen generator which produced hydrogen from the electrolysis of water. To combat the non-linear response of the FID, all gases (except THC) were oxidized to CO_2 then reduced to methane before introduction into the FID. Since all material that reached the FID was methane, the non-linear response from larger organic molecules was eliminated. Due to problems getting the entire cycle to work properly on the instrument in the AEDAS, the chromatograph was seldom used in that laboratory. Occasionally, the total hydrocarbon cycle could be activated properly in the AEDAS, so only THC data were collected on the south side of the roadway.

On the north side of the roadway, the entire cycle was normally activated except when time was not available to completely calibrate the instrument. The chromatograph used a 5Å molecular sieve to separate CO from CH_4. As this sieve became saturated with water, its separation ability was drastically decreased. The high humidities present in Houston would saturate the column rather quickly and thus distort the CO peak. Therefore, the CO data from this analyzer was often questionable. Since another CO monitor was used to draw samples off the same manifold as the Byron, project personnel usually neglected this problem. All other data from this instrument appeared to be accurate. The gas chromatographs had an accuracy of $\pm 1\%$ full scale or ± 0.1 ppm for all pollutants except CO_2 which had an accuracy of ± 10 ppm.

D. Carbon Monoxide Sensors

Carbon monoxide concentrations were measured using a Byron 401 gas chromatograph (discussed above), two DASIBI Model 3003 CO analyzers (the EPA designated standard), and up to seven Energetics Science Model 2600 Ecolyzers. Since this was the primary purpose of the project, a large amount of time was spent in an effort to get CO data.

One DASIBI CO monitor was located in MEL-2 and the other in AEDAS-1. The monitors drew samples from the glass manifolds in the laboratories (Figure 36). Since CO is a very stable gas, these air samples were the last to be drawn off the manifold. These analyzers used the technique of Gas Filter Correlation. Broadband infrared radiation was passed through two gas cells of a rotating gas filter wheel. One cell was filled with nitrogen and the other with carbon monoxide. Light emitting from the N_2 cell was CO sensitive while light emitting from the CO cell was optically scrubbed of the CO absorption wavelength. The two alternating beams emerging from the wheel were reflected many times across a multipass optical chamber where the sample gas resided. If no CO was present in the sample, no attenuation of the light occurred from either gas cell, or if attenuation occurred due to other species, it was equal on both sides of the wheel and had a net effect of zero. If CO was present in the sample, the radiation emerging from the CO cell would experience no further

attenuation; however, radiation from the N_2 cell would be attenuated by the CO in the sample. This attenuation created an imbalance of light in the alternating signals reaching the optical detector which was proportional to the CO concentration in the optical chamber. The stated accuracy of these monitors was ± 0.1 ppm. They had no expendable cells, were easily calibrated, and were probably the best monitoring instrument used on the project.

Energetics Science Ecolyzers were used for the majority of the CO monitoring. Depending upon the wind direction four or five units were placed on the north side of the roadway and the remaining units that were operational placed on the south side. These analyzers used an electrochemical acid cell to determine the CO concentration. The reaction occurred at a catalytically active platinum electrode according to the following equation:

$$CO + H_2O \longrightarrow CO_2 + 2H^+ + 2e^- \qquad (5-2)$$

The instruments had an accuracy of ± 0.5 ppm. They were easily operated but required frequent calibration to overcome the span and zero drifts. The accuracy of these instruments was also affected by the pH of the acid in the cell. As the cell aged, the acidity of the cell decreased and the accuracy of the analyzer also decreased. With careful attention and frequent calibration (about every 2 hours) these instruments had an error of no greater than 1 ppm of carbon monoxide.

If the wind was southerly, five units were used on the north side of the freeway to monitor carbon monoxide concentration. These instruments sampled air at heights of 102, 74, 47, 33, and 5 feet. The samples were taken from the same heights as the meteorological instruments were located. Samples from heights of 55 and 5 feet were usually taken on the south side of the road during prevailing southerly winds. When the winds were out of the north, the top Ecolyzer from the north tower was usually moved to the south side of the roadway and monitored at 37 feet.

Samples were drawn from heights over 5 feet using 1 inch diameter black polyethylene tubing that was allowed to weather several days before its use. Air was pulled through these tubes by vacuum pumps located downstream of the withdrawal point for the Ecolyzers. Since the Ecolyzers were quite sensitive to sample gas flow rates, the pressure in the vacuum lines was adjusted so that the vacuum never exceeded about 1 inch of water below atmospheric pressure. At this pressure, the air had a residence time of about $5\frac{1}{2}$ seconds for the highest receptor. (The air velocity was about 17.5 ft/sec.)

V. Tracer Gas Studies

Tracer gas studies were performed at the Houston research site on December 18, 1984, and December 19, 1984. These were the last two days of data collection for project 283. The studies utilized a rare, inert, non-toxic tracer gas to measure the dispersion in the atmosphere. The tracer gas chosen for the study was sulfur hexafluoride (SF_6).

The procedure involved the emission of a known flow rate of gas into the exhaust system of two vehicles that were constantly driven around the perimeter of the research area. The vehicles were paced so that they were at opposite sides of the research area at all times. The vehicles would enter the freeway, exit at N. Main St. or Airline Dr., make the U-turn underneath the freeway, and enter the freeway again traveling in the opposite direction. In order to eliminate the need for stopping at traffic lights at N. Main or Airline, the drivers were cleared through the intersection by Houston police officers. Therefore, the time spent driving on the freeway was maximized. A log was kept by passengers in each vehicle that included the number of trips around the research area, vehicle speed while passing the tower on the north side of the roadway, and the time that the vehicle passed the tower.

Air samples for SF_6 analysis were collected using Developmental Sciences syringe samplers. The samplers were donated by General Motors Corporation and had been modified by GM. They were further modified by TTI so that all samplers on the same tower were controlled by one timer. The timer would sequentially pull the six syringes on the sampler over a 15 minute interval. Therefore, the sampling lasted 90 minutes for each sampler.

The air samples were analyzed by a Valco Instruments gas chromatograph equipped with a Model 140B electron capture detector (ECD). The chromatograph was calibrated with 2.02 ppb SF_6 from Scott-Marrin. This gas was checked against a 2.0 ppb standard obtained from Matheson by Radian Corporation in Austin, Texas. A copy of the Radian report is included in Appendix E.

The flow rate of SF_6 emitted was measured using a Hastings electronic mass flowmeter. This instrument had an accuracy of $\pm 1\%$ full scale for air measurements. Since this instrument was calibrated for air, the heat capacity of SF_6 was used to convert the air flow rate to the appropriate SF_6 flow rate. The flow rate was measured before and after the experimental run.

Chapter 6

Data Processing

Two computers were used to monitor all instruments and store the data on magnetic tape. This section describes the data acquisition processes and the methods used to manipulate the collected data.

I. Radian DART

The Radian DART was used to monitor all instruments on the south side of the roadway except the UVW anemometers. This computer would collect the data, average the results, and output the averages to the user console. A nine-track tape drive equipped with an RS-232 interface was connected in parallel with the console. Therefore, all data that was transmitted to or from the console also appeared on the tape. Since the only information that was available from this computer were the calculated averages, no dynamic response could be determined from this data. However, this usually did not present a serious problem since most of the instruments connected to the DART were used to measure upwind concentrations. This did prevent, however, easy correction for calibration drifts.

There were other problems with the DART that had to be overcome. Often the Ecolyzers could not be interfaced well with the A/D on the DART because the computer would induce noise in the signal supplied by the Ecolyzer. Poor documentation on the software prevented the utilization of many features. Furthermore, the software was unable to properly average 360° wind vanes when the prevailing winds were northerly.

Since all data that appeared on tape from the DART were already averaged, the data only had to be printed on paper. As data were being sent to the tape drive, they were stored in an internal 2K byte buffer. When this buffer was filled, it was written on the tape. The tape drive was hardware selected so that the data were recorded in EBCDIC. Since the computer that eventually processed the data from the DART did not use EBCDIC, the data had to be converted to ASCII before printing.

In order to establish a drift factor for the instruments in the AEDAS, a special function of the DART software was utilized. When an instrument was being zeroed or spanned, the operator would key in a special code to the DART which caused the computer to begin printing instantaneous values at a specified frequency. Therefore, drifts in the calibration could be monitored. This procedure did not however, allow for automated correction due to drifts. The instruments interfaced to the DART were presented in Table 18.

II. Balcones Computer

The Balcones Computer software was designed so that instantaneous readings were recorded

on magnetic tape. While this method allowed for the realization of the full dynamic response of all data, it also meant that the data were recorded at a prodigious rate—over 160K of data an hour were recorded. The data collected by this computer had to be further averaged by user written software before it could supply useful information.

The Balcones computer was responsible for monitoring all instruments north of the freeway, all the radar units, and the UVW anemometers on the south side of IH610. The instrumentation for this computer was listed in Table 17. The computer actually sampled all of its 64 analog channels at 0.01 seconds; however, data were only retained for the selected sample frequency for each channel. As samples were being taken, the computer compared the values to a data file that contained the maximum and minimum permissible values for each channel. If a reading fell outside of this range, the operator was noted by a warning message at the console. The operator could then take the appropriate action or enter a message into the terminal which would appear on the tape.

The data were stored on nine-track magnetic tape in blocks of undefined length. The data were written in two byte (16 bit), two's complement form. Since the A/D used with the computer was a 12 bit converter and the A/D was selected for -5 to $+5$ VDC operation, a voltage of -5V was represented by 0 A/D counts and a voltage of $+5$V was represented by 4095 A/D counts. In order for the A/D counts to more realistically represent negative voltages, the value read from the converter was immediately added to -2047 before any further processing was done. Hence, negative voltages would then be represented by negative integers. Since the data were stored in two's complement form, a negative value was easily recognized by the value of the highest order bit in the 16 bit words. In order to facilitate the transferring of these words to tape, the data were placed in two temporary buffers in the memory of the computer. The size of these buffers was about 4K bytes. As data were being collected, a buffer would fill and then be written on the tape. As this buffer was being written, data would continue to be collected and stored in the second buffer.

The data inside each block on the tape consisted of several unseparated records. Each record had to meet a rigid format so that data reduction software could process the information. Each sample on the record had an associated record type which was used to classify the data stored in the records. A list of the various record formats is given in Table 19 and the record types for each channel were listed in Table 17. Record types 1 and 2 were used to indicate that a particular channel was in the calibration mode and record type 0 was used to indicate that the data for that particular channel was invalid.

Table 20 lists the channels which were calibration channels. All of these channels except the Byron (record type 5) had a zero and a span associated with the channel. The Byron only had a span value. When the time came for calibration of an instrument, the operator would issue a record type change to 0 for the instrument. The instrument would then be spanned and a record

Table 19

Record Formats Generated by the Balcones Computer

Each Row Represents the Contents of the Two-Byte Word Contained in that Location

Console Messages	Radar Records	Regular Channel Records
Length	Length	Length
ASCII Spaces	Channel	Channel
Time High	Name1	Time High
Time Low	Name2	Time Low
CO	Name3	Name1
MM	Name4	Name2
EN	Time High	Name3
T	Time Low	Name4
Type	Type	Type
ASCII Code	Cat 1 Count	Sample1
	Cat 1 Speed High	Type
	Cat 1 Speed Low	Sample2
•	Cat 2 Count	Type
	Cat 2 Speed High	Sample3
	Cat 2 Speed Low	
•		•
	•	•
	•	•
•	•	
		Type
	Cat 5 Count	Sample6
•	Cat 5 Speed High	
	Cat 5 Speed Low	
ASCII Code		

type of 2 was issued for the channel so that the span value could be appropriately marked on the tape. After the span was read for enough time to obtain an adequate sample count, the record type was returned to zero and the instrument was then zeroed. After the instrument reached its zero value, a record type of 1 was issued for the channel. Again, the zero was read for enough time so that an adequate count was available for averaging. The record type was then changed back to 0 and the analyzer allowed to monitor air. After the instrument stabilized on ambient air, the record type was returned to the type designated for the instrument. This same procedure was followed for the Byron except a record type of 1 for that instrument was not defined. With this procedure, linear span and zero drift factors could be applied in the data reduction phase.

In addition to writing the raw data to tape and reporting channel overflows, the Balcones also computed approximate 5 minute averages for all the channels. This enabled project personnel to monitor the data while it was being collected. If any data were suspicious, the operator could enter a console message and correct the problem.

III. Balcones Raw Data Reduction

The Hewlett-Packard HP-9000 series 500 computer at the Department of Chemical Engineering at Texas A&M University was used for data manipulation. Although the data resided on nine-track tape, it was written in raw binary form which the HP was not equipped to readily process. Furthermore, this type of data is not printable so that it could be inspected for errors.

The first step in the data processing phase involved the separation of the records internal to each tape block. This involves the insertion of delimiters between each record. Even though each record in the raw data contained a time stamp corresponding to the time that the first sample in the record was taken, no date stamp was included in the data. Therefore, a Julian date stamp was added to each record. The program that accomplished this was called SETA and is included in Appendix A. In addition to the above, the program would list the console messages on paper, trap errors that would cause subsequent data reduction programs to fail, (*e.g.*, a record type that was incompatible for a channel), and change the two ASCII spaces following the record length in the console message records to ASCII nulls. The output from this program was printed in hexadecimal form so that it could be inspected for errors.

The next step in the data processing phase included the editing of incorrectly recorded data. Examples of this type of data would include changing incorrectly entered record types that occurred during calibration. The data were not actually changed; only the record types that marked the data were changed. Additionally, all data known to be incorrect due to instrument failure were invalidated.

After the data were inspected, the UNIX sort utility was used to group the records so that subsequent data reduction was possible. The sort routine was keyed in the following order: year,

Table 20

Calibration Channels on the Balcones Computer

Channel	Instrument	Normal Record Type	Span and Zero?
16	TECO NO_x	4	Yes
20	TECO NO_2	4	Yes
22	TECO NO	4	Yes
24	Byron THC	5	No
26	Byron NMHC	5	No
28	Byron CH_4	5	No
30	Byron CO	5	No
32	Byron CO_2	5	No
49	Ecolyzer-1	17	Yes
51	Ecolyzer-2	17	Yes
53	Ecolyzer-3	17	Yes
55	Ecolyzer-4	17	Yes
57	Ecolyzer-5	17	Yes

day, channel, and time. Following the sort, the calibration data were averaged, then grouped, so that all start and end calibration data for a period appeared on a single record. Subsequently, the sort utility was again employed on the calibration data so that it could be used as an input file to the next data reduction program. The sort was keyed for this step in the following sequence: year, day, time, and channel. The programs used to perform this step of the data processing were collectively known as SETB and are included in Appendix B.

The third step in the data processing phase was to average the output from the SETB programs. Averages for five and fifteen minute periods as well as hourly averages and daily maximums were tabulated. Standard deviations for all instruments except radars were also reported. Calibration drift factors were applied in this stage. Vector sums for the UVW anemometers were performed that included the calculation of the wind azimuth and elevation angles. Non-cosine response factors were applied to the UVW data. The data were then written back to tape for permanent storage by a user written routine that generated IBM compatible tapes. This averaging program was known as SETC and is included in Appendix C.

The last step of the data reduction phase involved the storing of the SETB data in card image format. Discrete integer samples were converted to their floating point equivalents in the units of the respective instruments. As with the SETC programs, calibration drift factors and non-cosine response factors were applied to the discrete data points. This step was accomplished using the SETD program listed in Appendix D. All data were grouped by date and channel and placed in chronological order on magnetic tape. Before this program was run, the calibration data from the SETB program were resorted on the following keys: year, day, channel, and time. Therefore, SETD output contains discrete data points (corrected as described) at the selected instrument sample rates.

Chapter 7

Discussion of Results

The discussion of results from both model revision and experimental data collection will be discussed in this chapter. In the interest of brevity, only a comparison of the revised models to other existing data bases and models will be presented in this report. The reader that requires a more detailed treatment is referred to two other reports.[47,48]

I. TXLINE-2

The modifications to the original TXLINE model described in Chapter 3, give the model more flexibility for modeling the dispersion process near roadways. The model was converted from an infinite line source model to a finite line source model. This enabled the prediction of pollutant concentrations upwind of a roadway. More importantly, this modification gives the user the option to use the model for curved roads and other types of scenarios in which an infinite line source model would be inappropriate. The program is capable of modeling a single finite line source or several parallel finite line sources of any elevation. The model is primarily intended for use in predicting carbon monoxide concentrations, but can also be used to simulate the dispersion of other gaseous, non-reactive, pollutants.

The models TXLINE, TXLINE-2, CALINE3, and HIWAY-2 were compared to the experimental data bases discussed in Chapter 2. The statistics used for the comparison were the average error, the average squared error, the slope and intercept of a linear regression analysis, the regression coefficient, percent within 2 ppm (2 ppb SF_6), and the percent within 1 ppm (1 ppb SF_6).

A. Comparison to GM Data

The input parameters for the GM data base comparison, except for surface roughness, were obtained from the final report by Cadle.[35] A surface roughness of 0.30 meters was used in all models except HIWAY-2, which does not require this parameter. A 4.4 meter reference wind speed height was used in the TXLINE models.

The statistical analysis of the models for the GM data base is presented in Table 21 and Figure 37. TXLINE-2 did not perform as well in this comparison as TXLINE. This may be attributed to the fact that the original TXLINE model used the GM data base as the primary data base for its development.

B. Comparison to Texas Data

The models were compared to at-grade Texas sites in Houston, San Antonio, and El Paso. The input data were taken from Rodden.[63] Statistical comparisons are presented in Tables 22–24 and Figures 38–40. Scatterplots of TXLINE-2 for each receptor location are presented in Figures 41–43

Table 21

Statistical Analysis of the TXLINE-2 Model with the GM Data Base

Statistic	TXLINE-2	TXLINE	HIWAY-2	CALINE3
Average Error (ppb)	−.16	−.10	−.26	.11
Average Squared Error (ppb^2)	.35	.12	.30	.32
Slope	.94	.93	.91	.69
Intercept (ppb)	.22	.17	.35	.19
R^2	.57	.80	.67	.47
Number of points	471	471	561	561
% within 2 ppb	99	100	99	NA
% within 1 ppb	91	97	93	NA

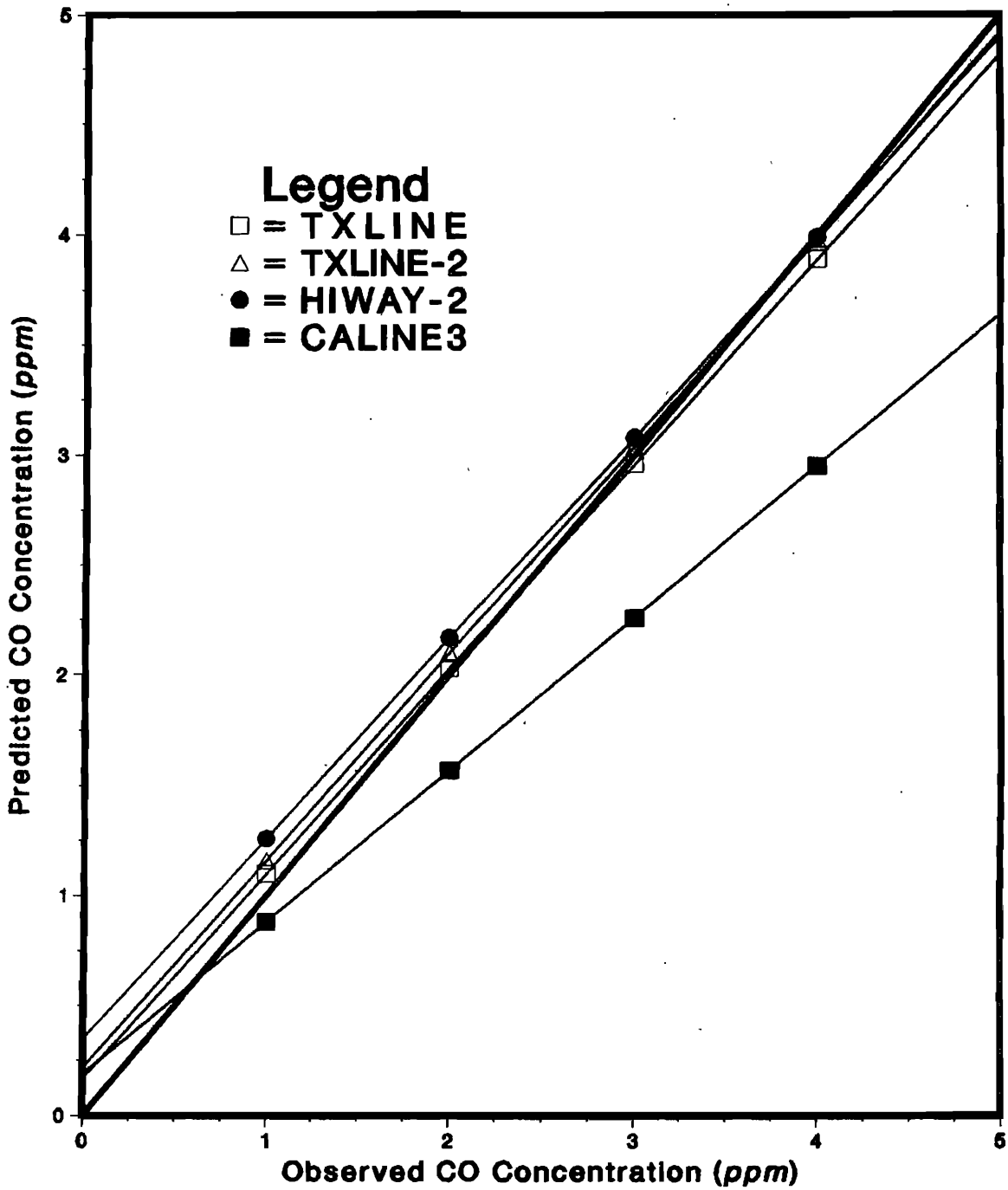


Figure 37

Regression Lines for the TXLINE-2 Model using the GM Data Base

for the El Paso data base. These scatterplots indicate that the ability to predict concentrations decreased as the distance from the roadway increases.

The Texas data comparison does not allow for a determination of model superiority or inferiority. However, the data did show that each model had a tendency to underpredict the concentration.

C. Comparison to SRI Data

The models were compared to the SRI data base both using elevated and at-grade data. These comparisons allow an unbiased test of all models except CALINE3 since these data were used in the development of the model.

The comparison to the SRI elevated data base enables an evaluation of modeling for elevated sources. Again, the input information was taken from Rodden.[63] The comparisons are illustrated in Table 25 and Figure 44 for the elevated SF_6 experiment. The comparisons for the SRI elevated carbon monoxide data are given in Table 26 and Figure 45. TXLINE-2 compared most favorably with this data set. All models gave similar values for the average error and average squared error. The TXLINE-2 model had the highest regression coefficient and slope, and the lowest intercept. All models again underpredicted the data.

Results from the SRI at-grade site are presented in Tables 27–28 and Figures 46–47 for SF_6 and CO, respectively. The SRI data base failed to clearly differentiate between the models. However, it appears that TXLINE-2 may have performed slightly better than the other models.

Table 22

Statistical Analysis of the TXLINE-2 Model

Using the San Antonio Data

Statistic	TXLINE-2	TXLINE	HIWAY-2	CALINE3
Average Error (ppm)	.01	.01	.03	.14
Average Squared Error (ppm^2)	1.25	1.02	1.08	1.05
Slope	.41	.46	.46	.47
Intercept (ppm)	.73	.67	.64	.52
R^2	.17	.25	.23	.25
Number of points[†]	352	352	352	352
% within 2 ppm	93	96	95	95
% within 1 ppm	61	66	68	71

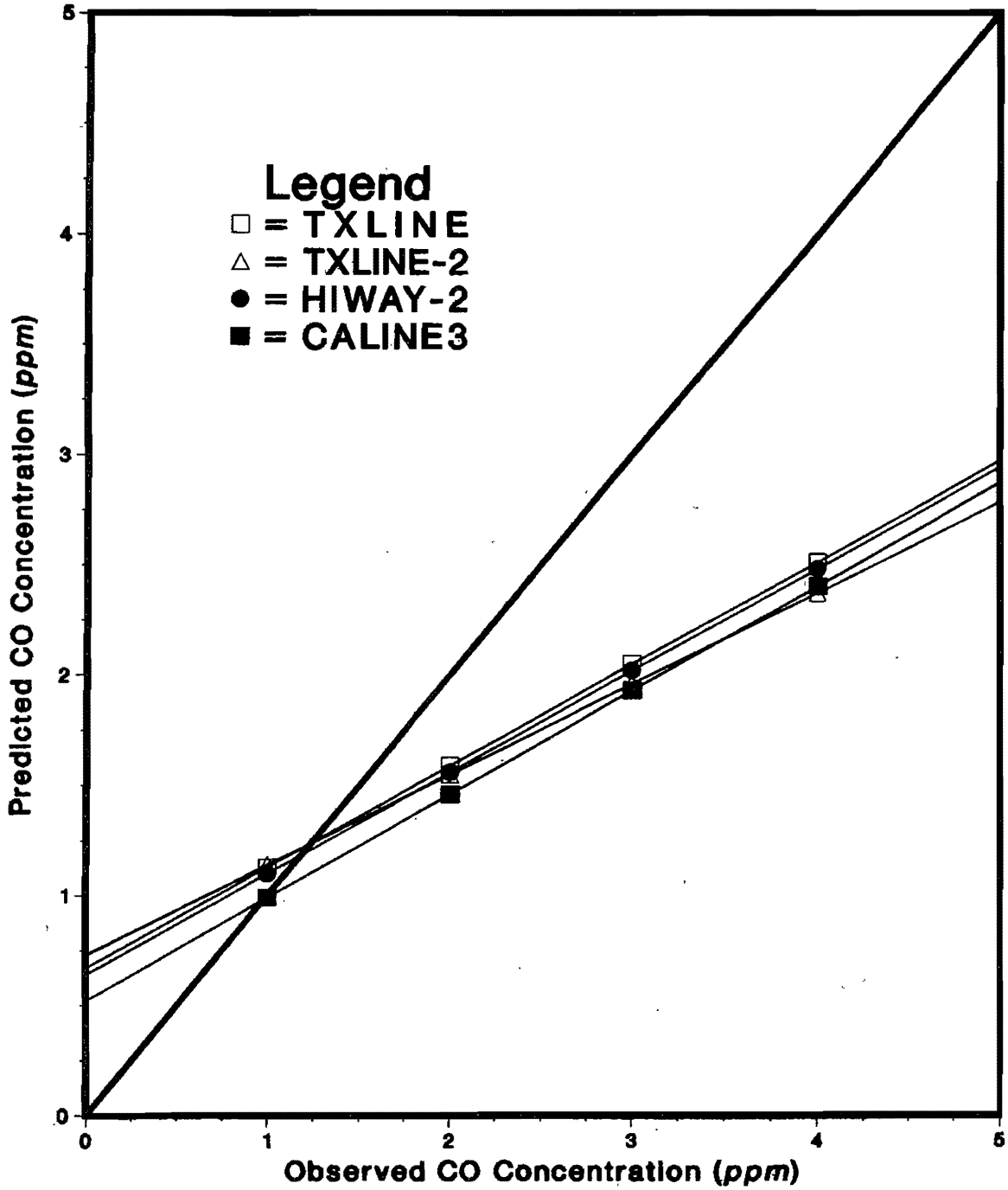


Figure 38

Regression Lines for the TXLINE-2 Model

Using the San Antonio Data

Table 23

Statistical Analysis of the TXLINE-2 Model Using the El Paso Data

Statistic	TXLINE-2	TXLINE	HIWAY-2	CALINE3
Average Error (ppm)	.36	.30	.25	.20
Average Squared Error (ppm^2)	3.05	3.01	2.82	2.65
Slope	.25	.22	.38	.33
Intercept (ppm)	.69	.78	.62	.58
R^2	.30	.31	.34	.37
Number of points[†]	704	704	704	704
% within 2 ppm	78	78	80	80
% within 1 ppm	46	44	50	47

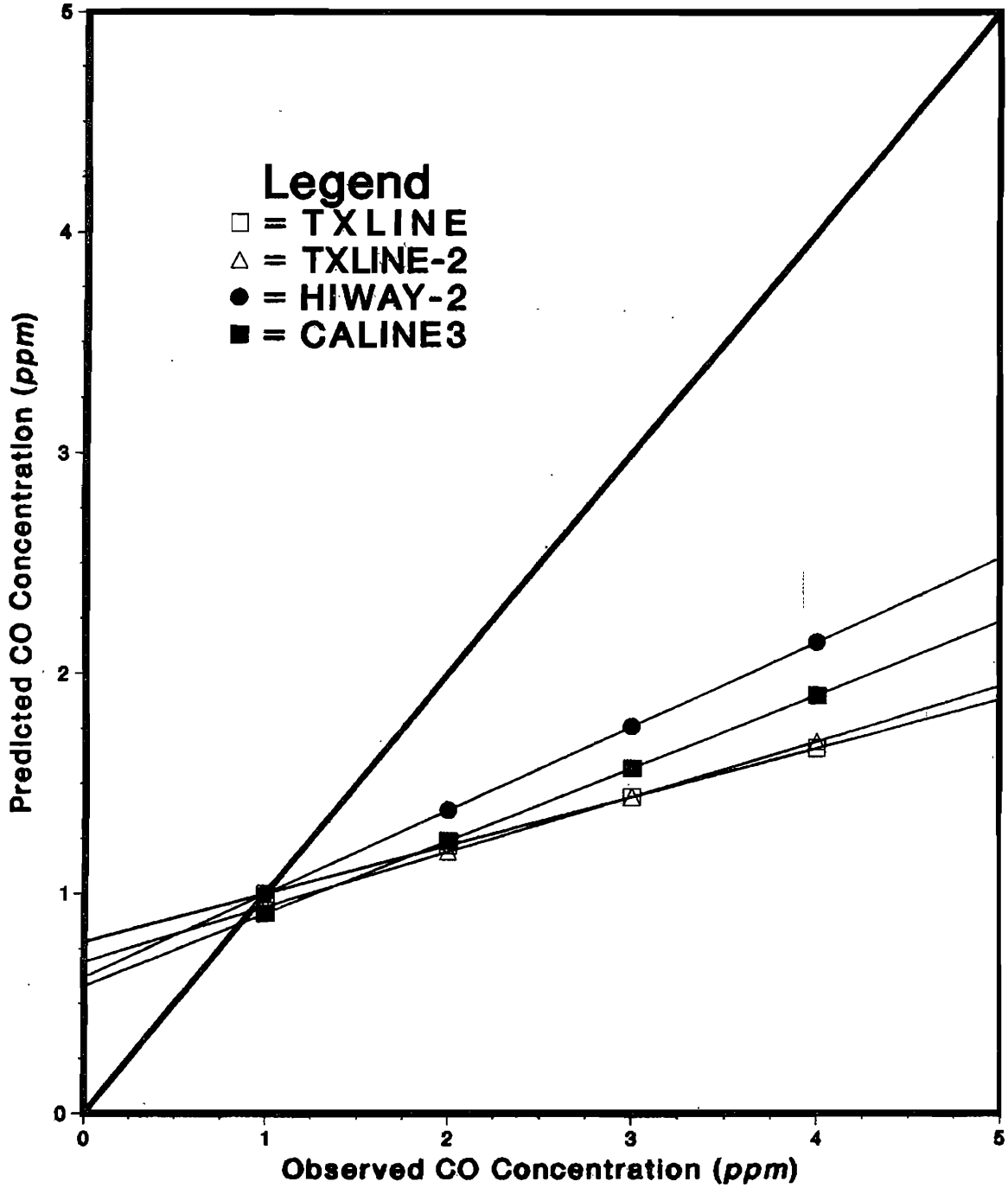


Figure 39

Regression Lines for the TXLINE-2 Model

Using the El Paso Data

Table 24

Statistical Analysis of the TXLINE-2 Model Using the Houston Data

Statistic	TXLINE-2	TXLINE	HIWAY-2	CALINE3
Average Error (ppm)	.35	.30	.42	.20
Average Squared Error (ppm^2)	1.03	1.03	1.21	1.19
Slope	.39	.34	.33	.48
Intercept (ppm)	.50	.61	.50	.52
R^2	.40	.39	.32	.32
Number of points[†]	195	195	195	195
% within 2 ppm	96	96	94	95
% within 1 ppm	64	67	64	59

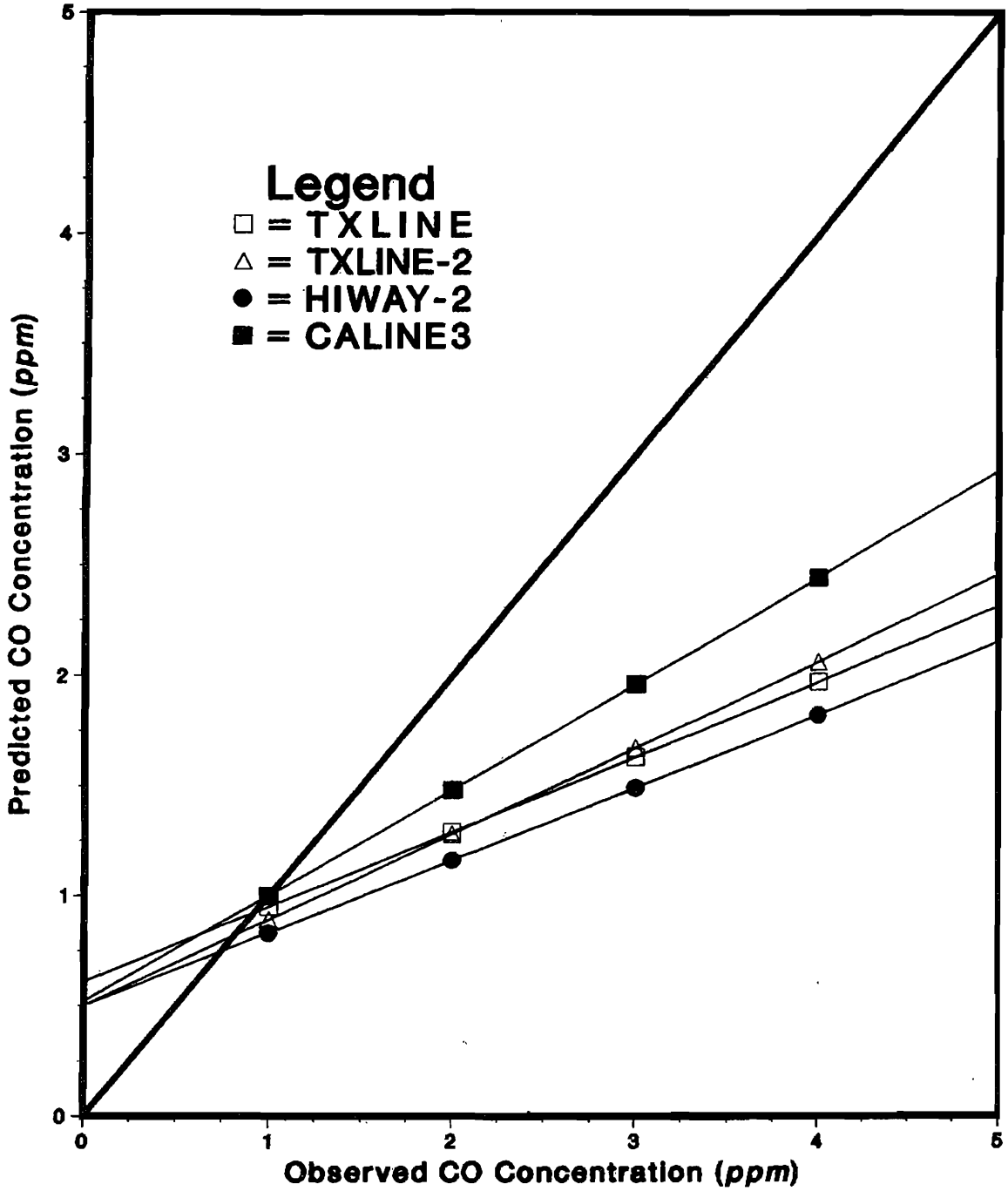


Figure 40

Regression Lines for the TXLINE-2 Model

Using the Houston Data

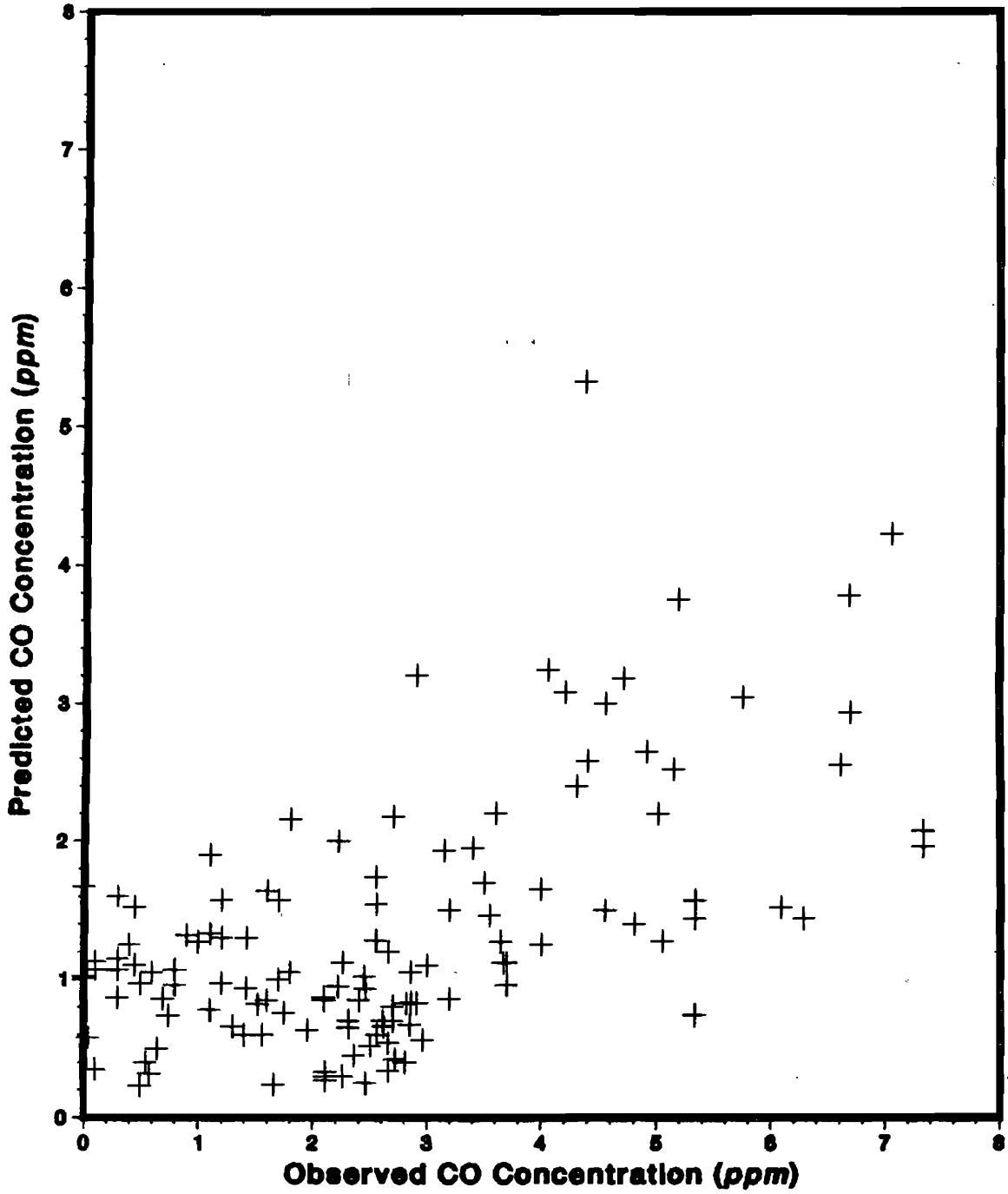


Figure 41

Scatterplot for TXLINE-2 Model at a 25.9 m Downwind Receptor

El Paso Data Base

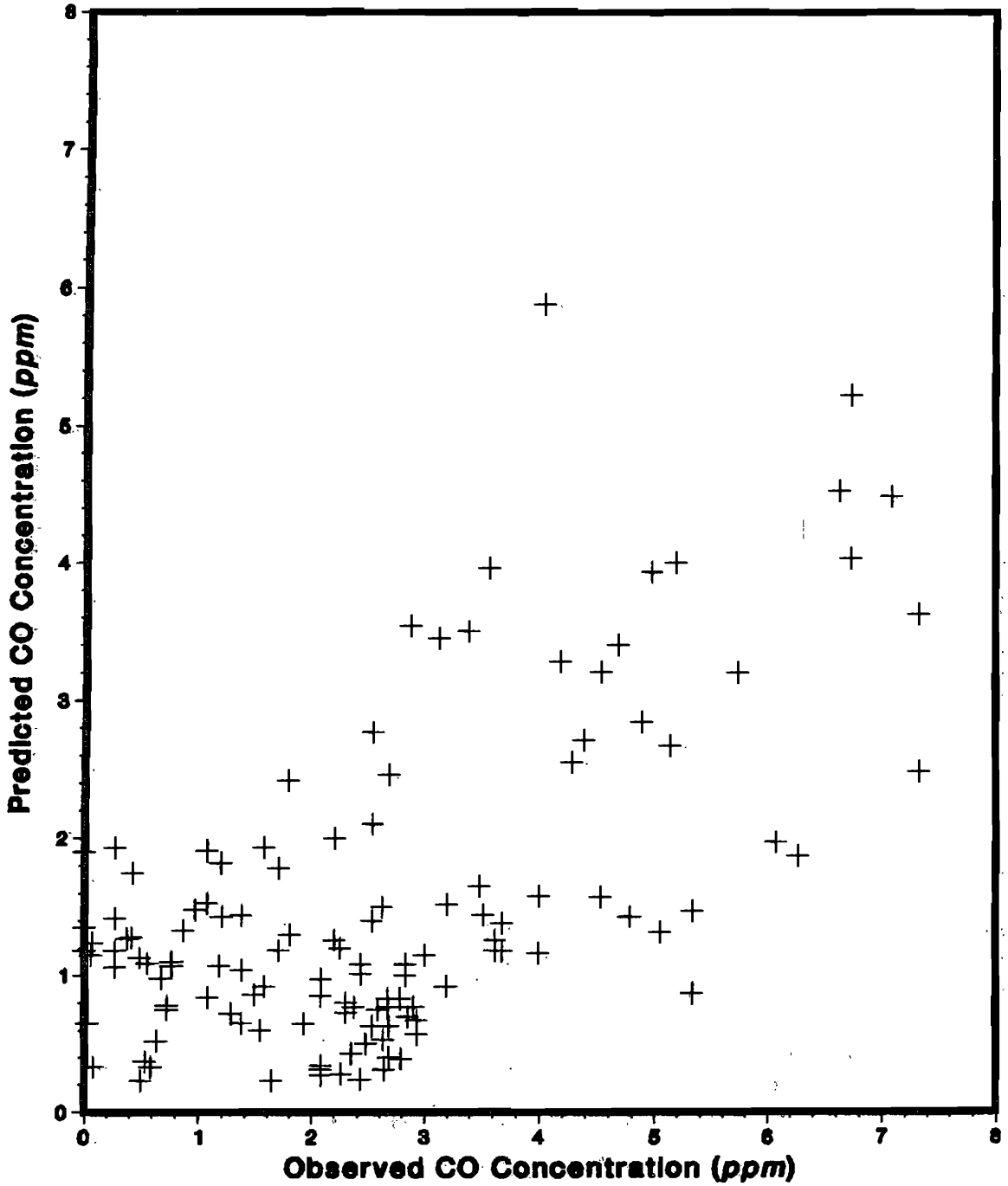


Figure 42

Scatterplot for TXLINE-2 Model at a 32.3 m Downwind Receptor

El Paso Data Base

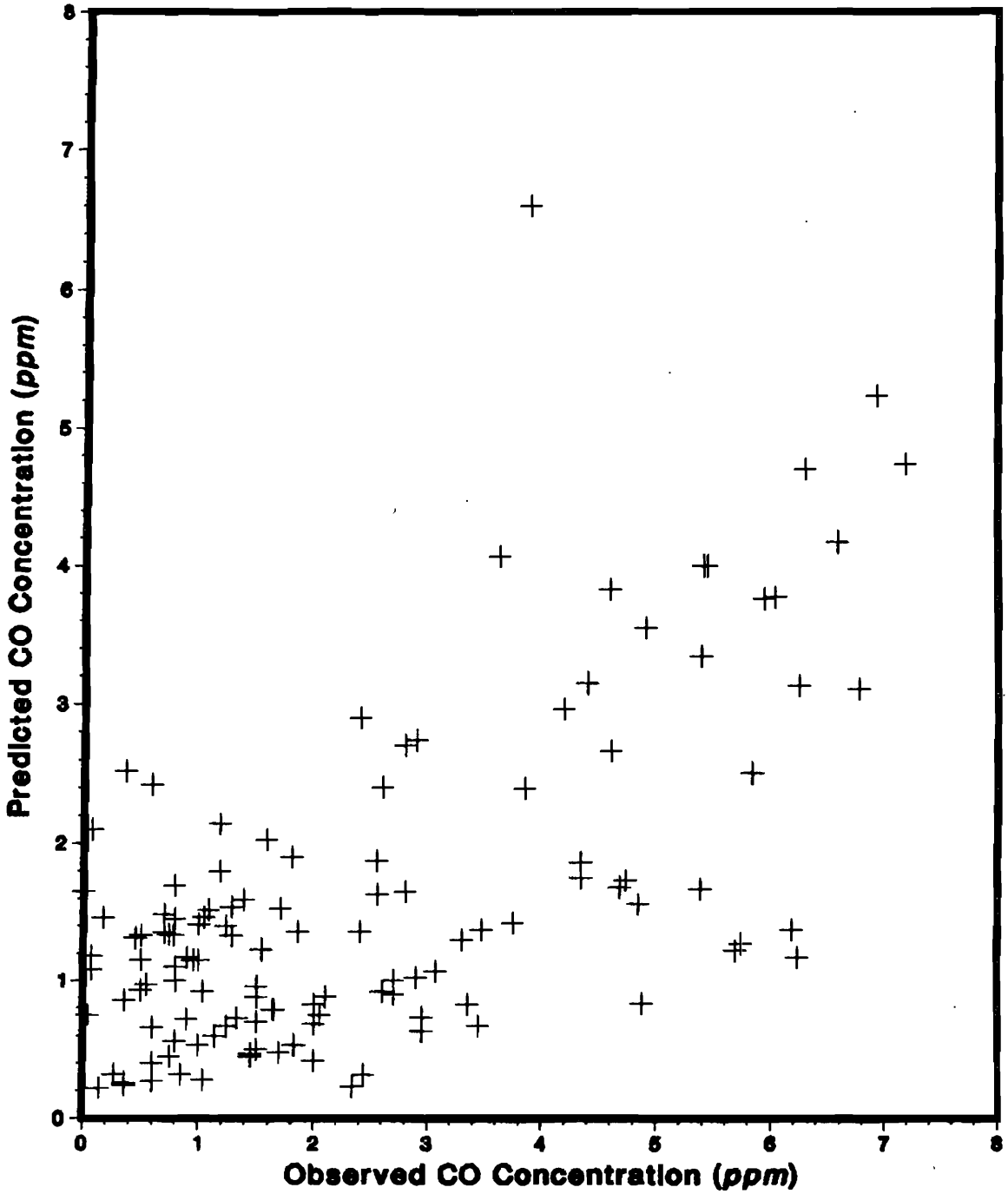


Figure 43

Scatterplot for TXLINE-2 Model at a 44.5 m Downwind Receptor

El Paso Data Base

Table 25

Statistical Analysis of the TXLINE-2 Model Using the SF_6 Data at the Elevated SRI Site

Statistic	TXLINE-2	TXLINE	HIWAY-2	CALINE3
Average Error (ppb)	.15	.41	−.09	.32
Average Squared Error (ppb^2)	4.83	4.62	10.62	3.87
Slope	.23	.09	.22	.23
Intercept (ppb)	.78	.94	1.24	.82
R^2	.06	.04	.03	.16
Number of points	336	336	336	336
% within 2 ppb	81	82	80	82
% within 1 ppb	62	66	66	66

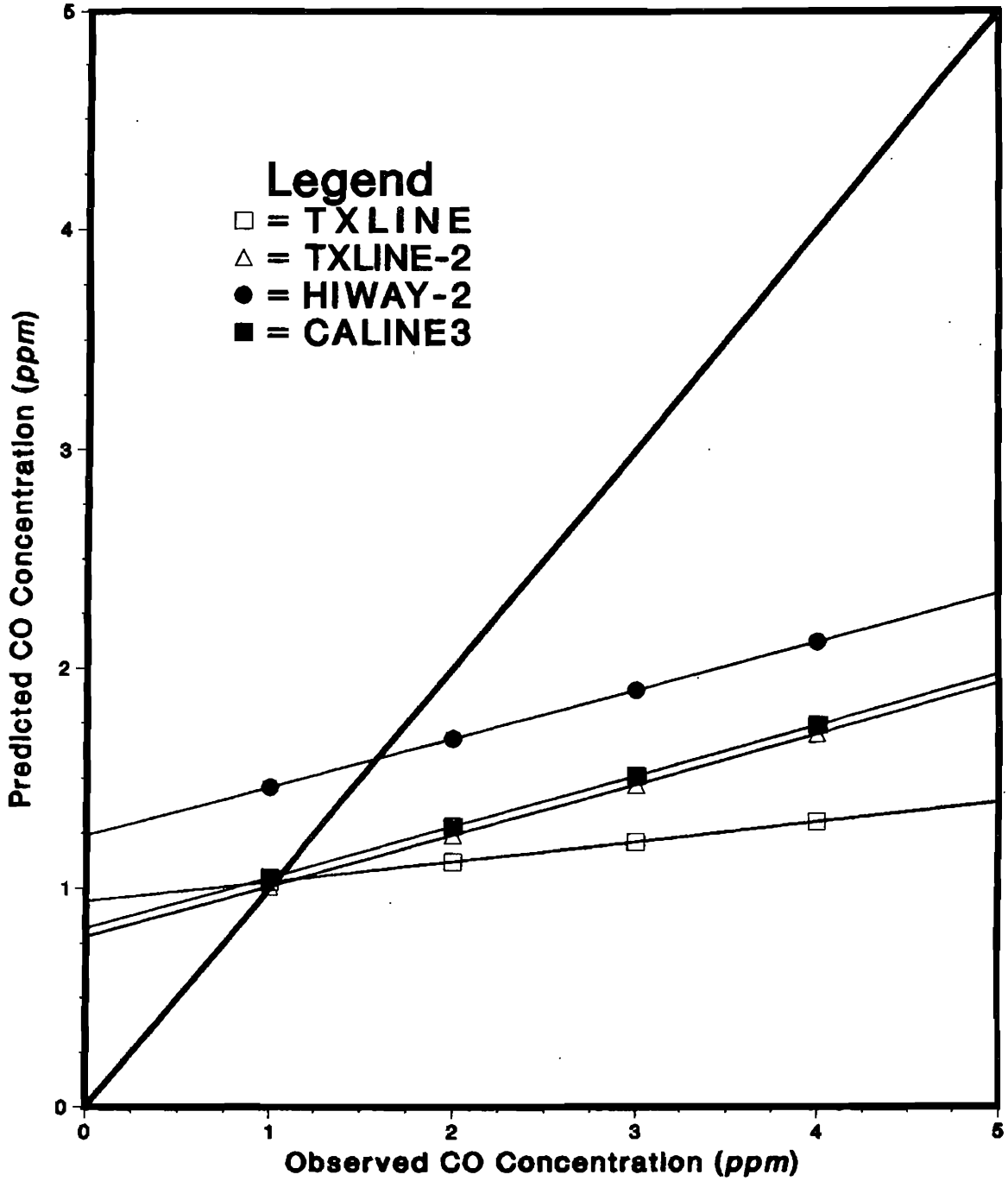


Figure 44

Regression Lines for the TXLINE-2 Model

Using the SF_6 Data at the Elevated SRI Site

Table 26

Statistical Analysis of the TXLINE-2 Model

Using the CO Data at the Elevated SRI Site

Statistic	TXLINE-2	TXLINE	HIWAY-2	CALINE3
Average Error (ppm)	.71	.67	.54	.67
Average Squared Error (ppm^2)	1.64	1.57	2.15	1.59
Slope	.22	.09	.13	.10
Intercept (ppm)	.20	.39	.47	.38
R^2	.13	.08	.02	.07
Number of points	359	359	359	359
% within 2 ppm	91	93	91	92
% within 1 ppm	66	70	70	68

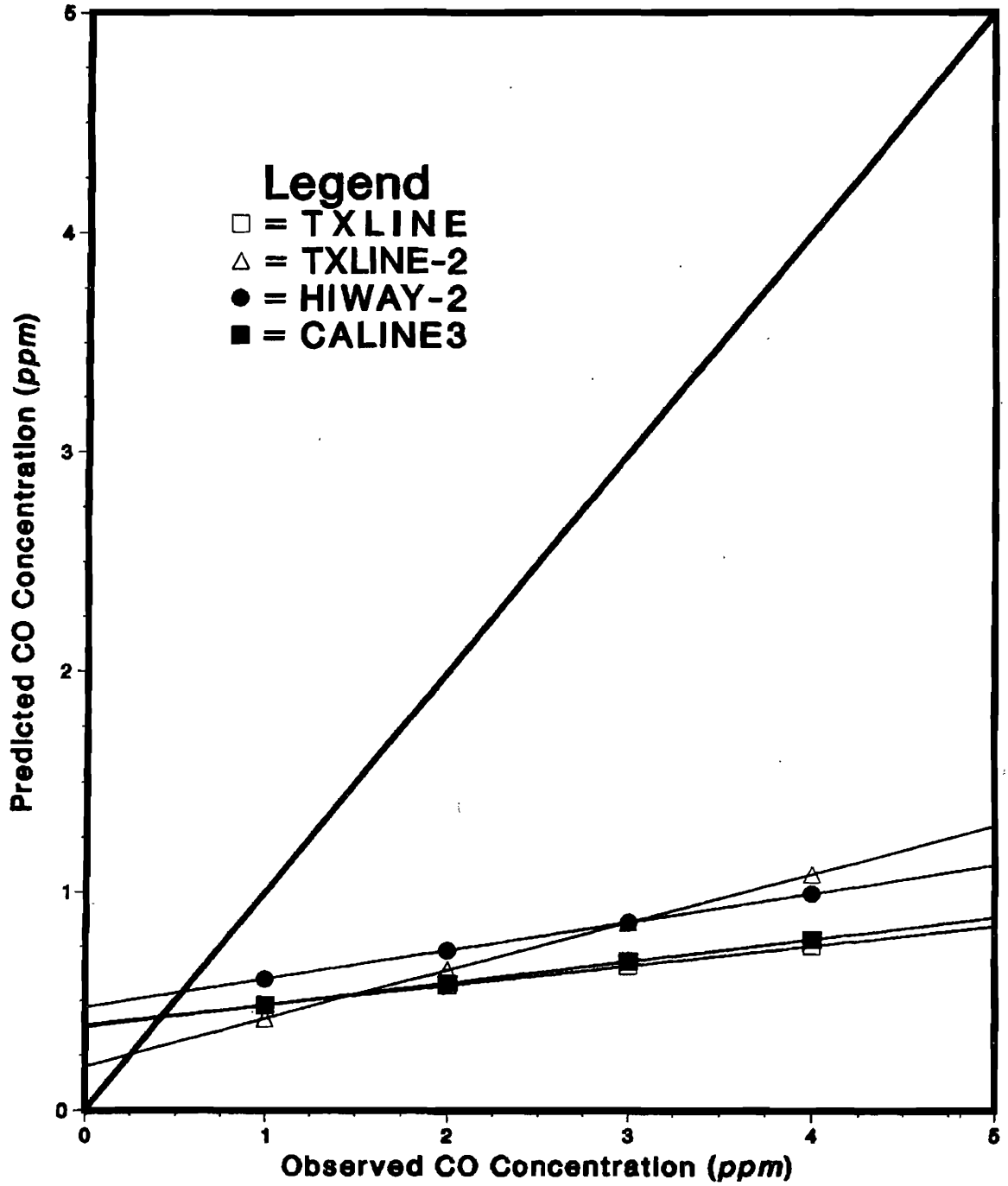


Figure 45

Regression Lines for the TXLINE-2 Model

Using the CO Data at the Elevated SRI Site

Table 27

Statistical Analysis of the TXLINE-2 Model

Using the SF_6 Data at the at-Grade SRI Site

Statistic	TXLINE-2	TXLINE	HIWAY-2	CALINE3
Average Error (ppb)	.68	.49	.73	.90
Average Squared Error (ppb^2)	15.45	19.23	19.82	21.30
Slope	.14	.10	.08	.06
Intercept (ppb)	.92	1.33	1.13	.99
R^2	.17	.09	.08	.04
Number of points	479	479	479	479
% within 2 ppb	80	85	81	76
% within 1 ppb	61	65	62	50

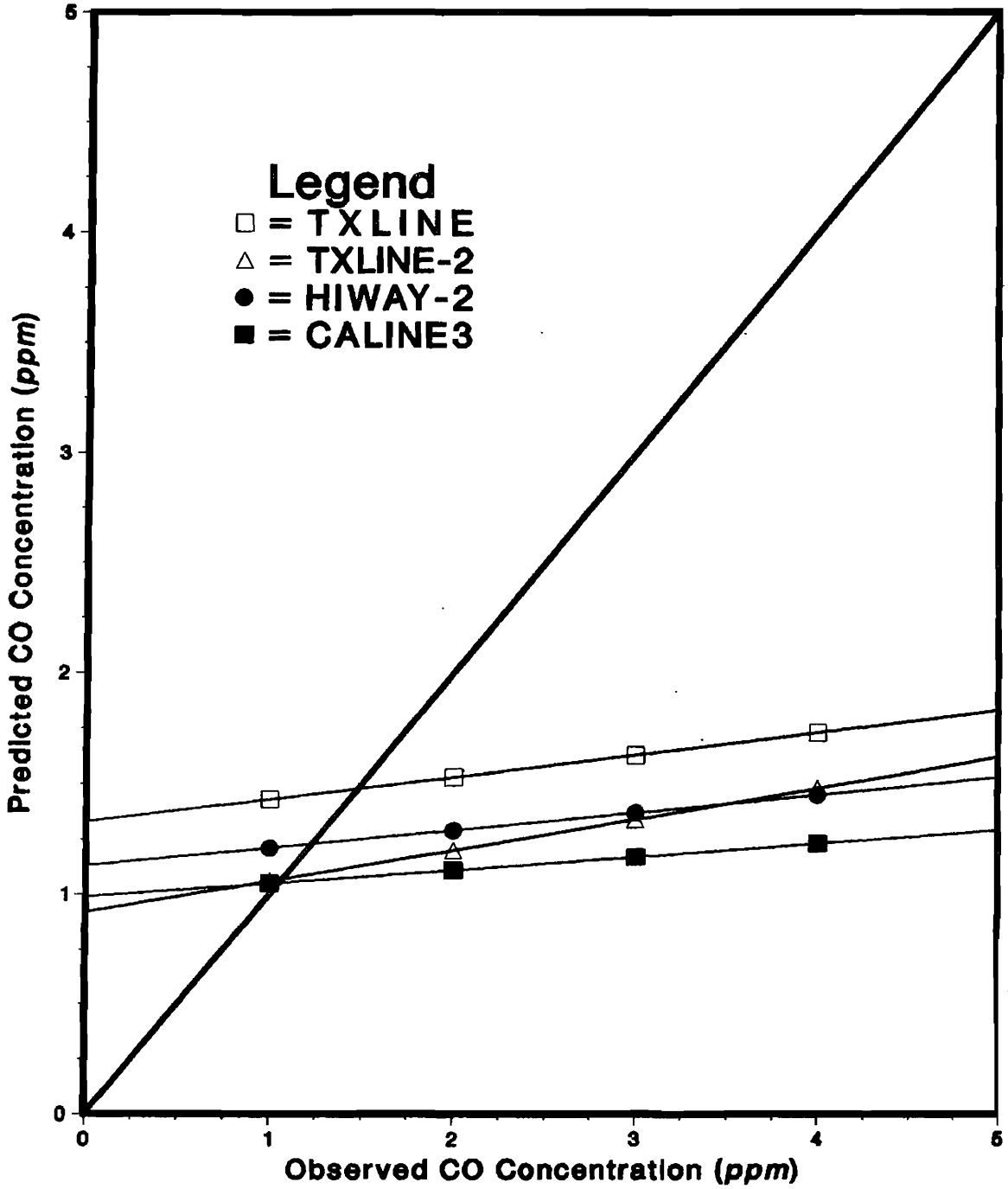


Figure 46

Regression Lines for the TXLINE-2 Model

Using the SF_6 Data at the at-Grade SRI Site

Table 28

Statistical Analysis of the TXLINE-2 Model

Using the CO Data at the at-Grade SRI Site

Statistic	TXLINE-2	TXLINE	HIWAY-2	CALINE3
Average Error (ppm)	−.07	.05	.30	.48
Average Squared Error (ppm^2)	3.63	1.54	1.68	2.80
Slope	.52	.47	.37	.24
Intercept (ppm)	.77	.73	.68	.65
R^2	.18	.37	.33	.10
Number of points	463	463	463	463
% within 2 ppm	86	93	91	84
% within 1 ppm	63	67	55	60

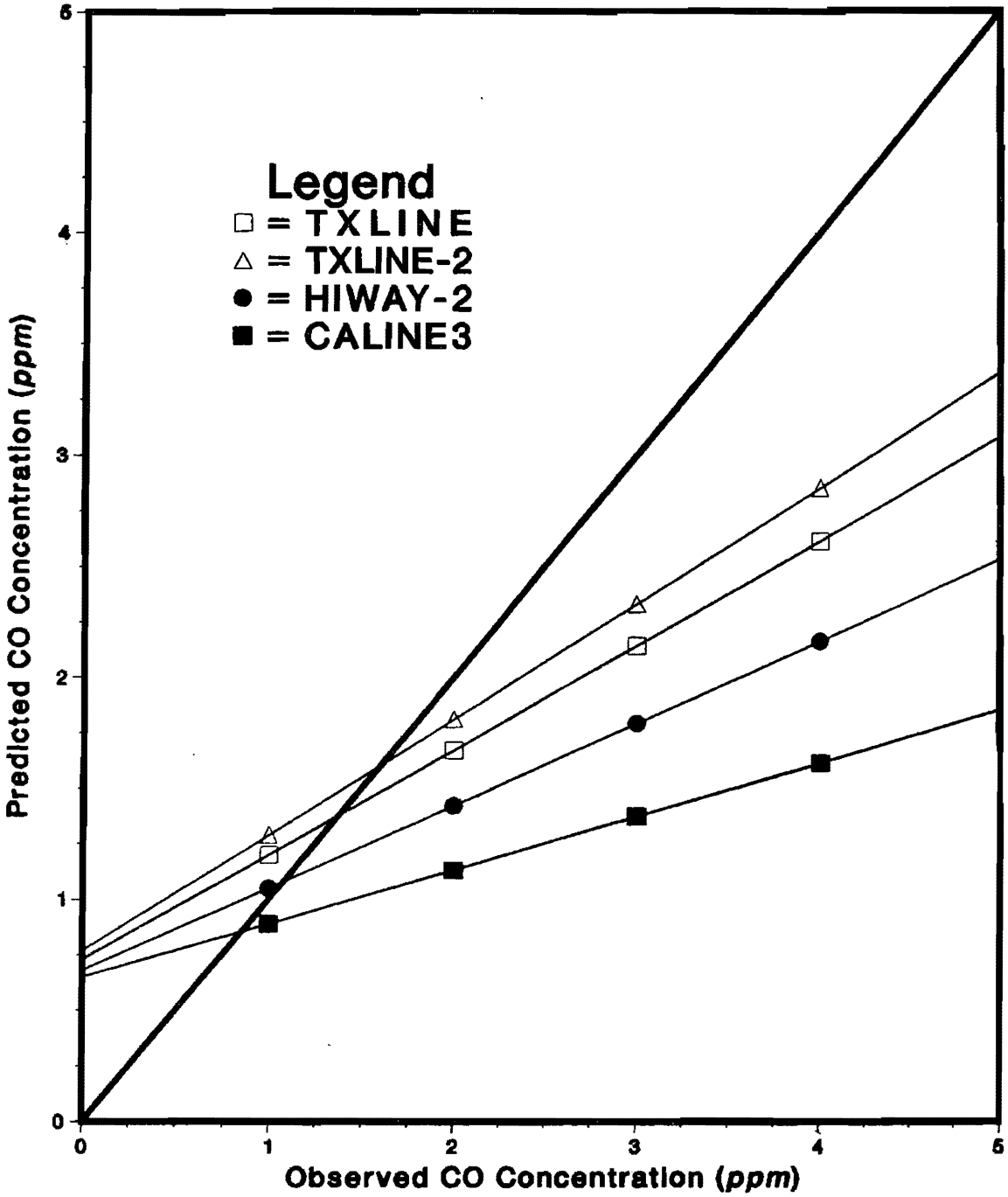


Figure 47

Regression Lines for the TXLINE-2 Model

Using the CO Data at the at-Grade SRI Site

II. TEXIN2

The original Texas Intersection Model (TEXIN) was revised as described in Chapter 3. The TEXIN model, previously developed to predict carbon monoxide concentrations near intersections, had several limitations which inhibited its use in a large number of cases. Many of these limitations have been resolved in the revised TEXIN2 model. Inclusion of a modified version of the EPA model MOBILE3 provides much flexibility in the estimation of emissions. The MOBILE3 routine allows for vehicle anti-tampering and inspection/maintenance programs to be simulated. A short cut emissions procedure was also developed for those users who do not desire to use MOBILE3. The analyst has many features in the new model which may be selected if needed for the intersection being modeled.

Validation of the revised TEXIN2 model was accomplished by a statistical comparison of the model predictions with three intersection data bases. Two of the data bases were collected in Texas while the third was collected in California. All of the data bases were discussed in Chapter 2. The statistics used in the comparisons included the slope and intercept of a linear regression analysis, average error, average squared error, percent within 2 ppm, and percent within 1 ppm.

A. Comparison to the College Station Data Base

The intersection models IMM, MICRO, TEXIN, and TEXIN2 were compared to the College Station data base discussed in Chapter 2. Different combinations of TEXIN2 user options were invoked in the process.

The stability class used in the model was determined using an analysis by Smith.[64] Parameters needed for the determination of atmospheric stability included insolation and wind speed which was taken as the average of the four anemometers used at the site over a 15 minute sampling period. Figure 48 gives the stability class curves for the model. The wind direction and ambient temperature were taken as the arithmetic mean of the wind vanes and thermometers, respectively.

Suggested surface roughnesses for various terrain are presented in Table 29. A surface roughness height of 1.5 meters was chosen in accord with the CALINE3 User's Guide. This height was chosen to account for the shopping center and single story houses in the southwest quadrant. Furthermore, the wind was predominantly southwesterly. A 1000 meter mixing height was used in all cases. Additional pertinent intersection data are presented in Table 30. All links had exclusive left-turn phases. Traffic volumes and speeds were estimated from loop counters present at the College Station site. Since the CMA Operations and Design procedure adjusts for lane widths, all lanes were set to the standard width of 12 ft. However, the width of each link is as presented in Table 30. Other required data including the VMT mix and the percent hot/cold starts were obtained from the Texas State Department of Highways and Public Transportation and are listed in Table 31.

The observed carbon monoxide concentrations were calculated as the measured downwind

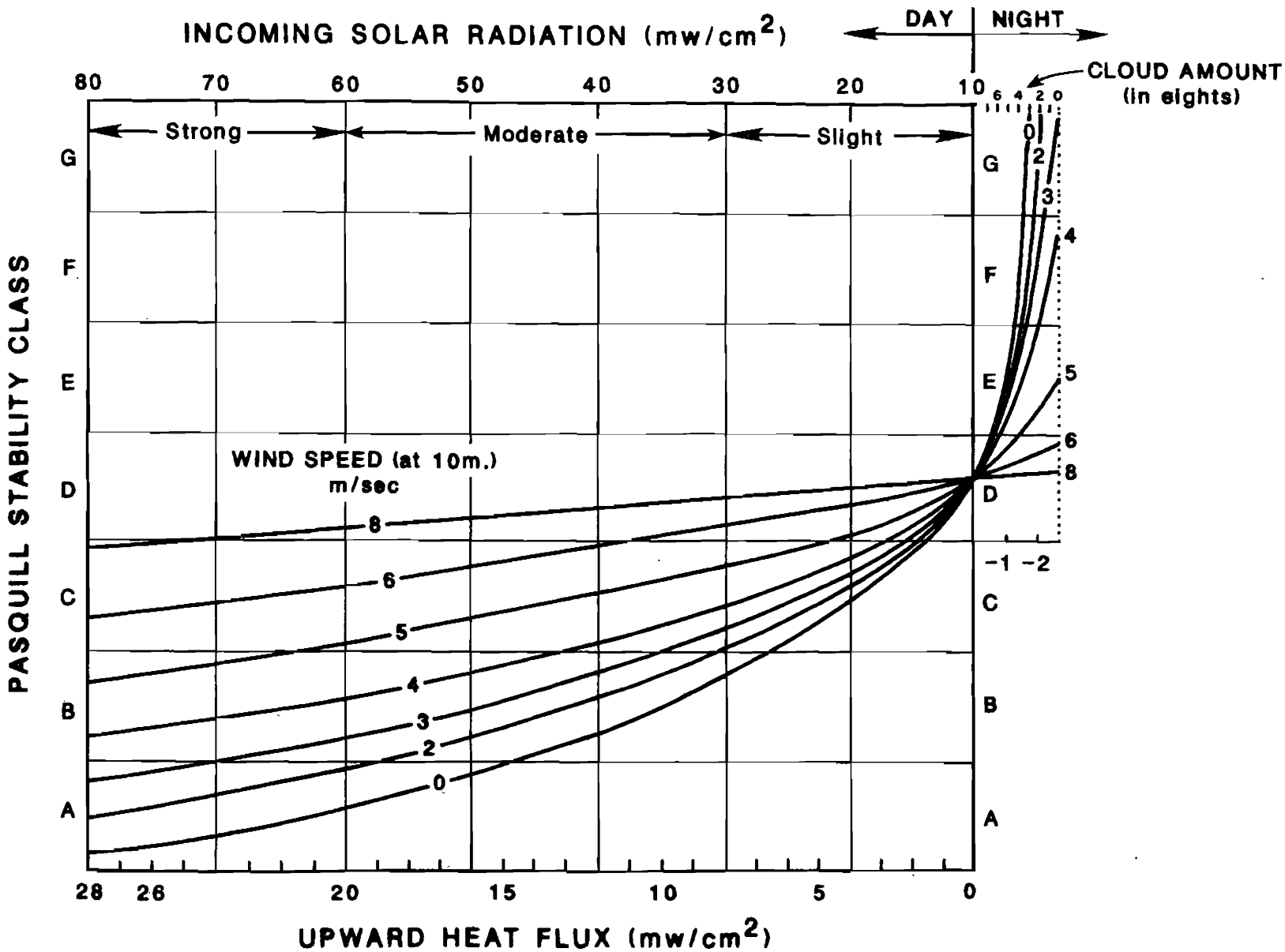


Figure 48

Stability Class Curves for the TEXIN2 Model[64]

Table 29

Surface Roughnesses for Various Types of Terrain[3]

Type of Surface	Roughness z_0 (cm)
Smooth mud flats	0.001
Tarmac (pavement)	0.002
Dry lake bed	0.003
Smooth desert	0.03
Grass	
(5–6 cm)	0.75
(4 cm)	0.14
Alfalfa (15.2 cm)	2.72
Grass	11.4
Wheat (60 cm)	22
Corn (220 cm)	74
Citrus orchard	198
Fir forest	283
City land-use:	
Single-family residential	108
Apartment residential	370
Office	175
Central-business district	321
Park	127

Table 30

Input Data for the College Station Statistical Analyses

Intersection Link Descriptions

Link	Length (m)	Width (m)	Height (m)	Number of Exclusive Lanes		
				Approach	Left Turn	Right Turn
North Texas	609.6	27.1	0	2	1	1
South Texas	400	27.1	0	2	2	0
Jersey Street	457.2	28.7	0	2	2	1
Kyle Street	259.1	17.1	0	2	1	0

Miscellaneous Data

Parameter	Value
Surface Roughness	150 cm
Averaging Time	15 min
Mixing Height	1000 m
Cycle Length	80 sec
Signal Phases	8
Number of Receptors	6

Table 31

Motor Vehicle Data Used in the College Station Statistical Analyses

VMT Mix for Brazos County, Texas

Year	LDGV[a]	LDGT1[b]	LDGT2[c]	HDGV[d]	LDDV[e]	LDDT[f]	HDDV[g]	MC[h]
1980	.590	.224	.108	.038	.000	.000	.036	.004

[a]Light Duty Gas Vehicles
[b]Light Duty Gas Trucks (GVWR < 6001 lbs)
[c]Light Duty Gas Trucks (GVWR < 8501 lbs)
[d]Heavy Duty Gas Vehicles
[e]Light Duty Diesel Vehicles
[f]Light Duty Diesel Trucks
[g]Heavy Duty Diesel Vehicles
[h]Motorcycles

Percent of Hot/Cold Starts for 1980 in Brazos County, Texas

Hour	PCCN[a]	PCHC[b]	PCCC[c]
00–02	44	11	54
03–05	54	7	59
06–08	46	13	57
09–11	24	32	52
12–14	18	33	37
15–17	27	27	39
18–20	16	29	40
21–23	27	18	48

[a]Percent VMT accumulated by cold start non-catalyst vehicles
[b]Percent VMT accumulated by hot start catalyst vehicles
[c]Percent VMT accumulated by cold start catalyst vehicles

Source: Texas State Department of Highways and Public Transportation

concentration minus the average measured upwind concentration. Since Tower 4 was the primary upwind tower, theoretically the only cases for which true background concentration values could be obtained were those with a wind out of the southwest quadrant (Figure 21). This includes cases in which the wind was blowing from any angle between the southern leg of Texas Avenue and the western leg of Jersey Street. For winds blowing from the northwest quadrant, emissions from vehicles on Jersey Street could alter the background value measured at Tower 4. However, most emissions are emitted near the intersection and Tower 4 was located at a considerable distance from Jersey Street. Consequently, it was assumed that for winds blowing from the northwest quadrant, measured concentrations at Tower 4 were valid background values for wind angles less than 45° (as measured from Jersey Street). For wind angles greater than 45°, the Tower 4 receptors were affected by vehicles encountering delay on Jersey Street and thus those data were omitted from the analyses.

The intersection models were statistically compared for 153, 15-minute cases of the College Station data base. These comparisons are summarized in Table 32 and Figure 49. The results show that TEXIN2 is somewhat better than TEXIN and the IMM, while MICRO exhibits the worst performance of the models.

A statistical review of TEXIN2 and TEXIN is summarized by Table 33 and Figure 50 for several of the various options available in TEXIN2. Each of the four runs improve at least five out of the seven statistics shown in the table.

Scattergrams of four trials of TEXIN2 with different user options as well as of the original TEXIN model are presented in Figures 51–55. The amount of scatter in the results is indicated by the magnitude of the average squared error. The statistical analyses for the College Station data base show that the TEXIN2 model will, on the average, predict more accurately than any of the other models compared. Many of the statistical parameters are nearly optimum. The regression line slopes are near unity and the regression line intercepts are near zero. The average predictive error is quite close to zero, especially for CMA Planning—MOBILE3 and CMA Operations and Design—Short Cut Method combinations. However, this better predictive capability is at the expense of an increase in scatter for this data base.

In order to fully analyze the predictive capabilities of the TEXIN2 model, the program was compared to the experimental data for various combinations of wind speed and wind angle. Three wind speed classes were chosen: low (0 to 2 m/sec), medium (2 to 4 m/sec), and high (above 4 m/sec); and three wind angle classes were chosen: near-parallel (0° to 30°) to the roadway, near 45° (30° to 60°) to the roadway, and near-perpendicular (60° to 90°) to the roadway. To avoid confusion, the wind/roadway angle was taken to be the wind angle with respect to the leg of the intersection that is the largest contributor of pollutant at the receptors. These above classes yield nine distinct wind speed/angle combinations. Scatterplots for these comparisons are presented in

Table 32

Intersection Model Statistical

Comparisons for the College Station Data

Statistic	TEXIN	TEXIN2†	IMM	MICRO
Slope	0.73 ± 0.04	0.93 ± 0.05	0.81 ± 0.04	0.23 ± 0.02
Intercept (ppm)	0.14 ± 0.10	0.12 ± 0.11	0.80 ± 0.10	0.26 ± 0.05
R^2	0.380	0.407	0.373	0.182
Avg. Error (ppm)	−0.36	−0.0008	0.47	−1.16
Avg. Sq. Error (ppm^2)	2.13	2.72	2.67	3.12
Number of Points	539	539	539	539
within 2 ppm	460 (85%)	463 (86%)	446 (83%)	418 (78%)
within 1 ppm	328 (61%)	345 (64%)	327 (61%)	277 (51%)

†Options used: CMA planning procedure with MOBILE3

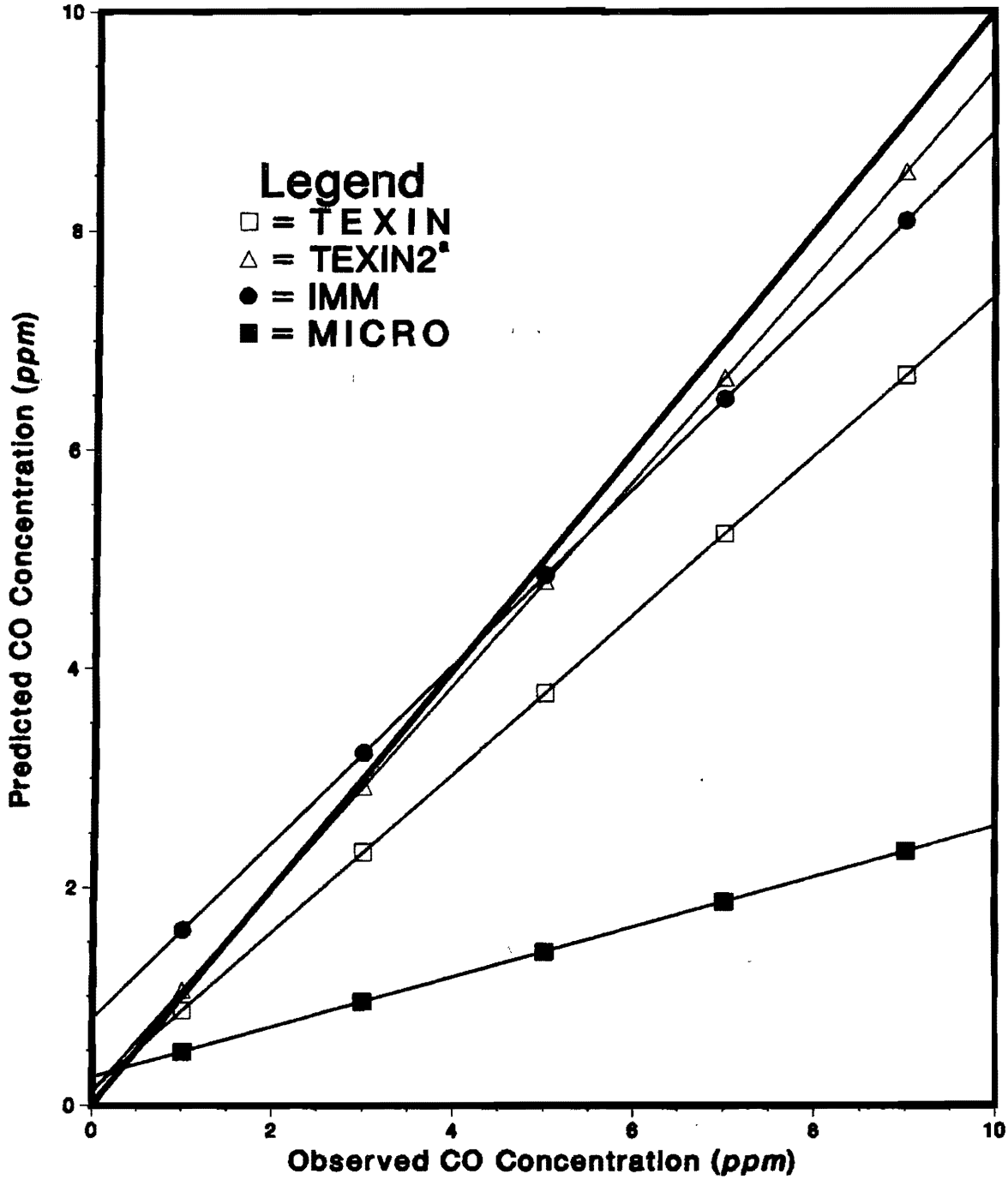


Figure 49

Regression Lines for Intersection Models

Using the College Station Data

[a]CMA Planning—MOBILE3

Table 33

TEXIN and TEXIN2 Model Results

A Comparison of the Various Options Available in TEXIN2

College Station Data Base

Statistic	TEXIN	TEXIN2[a]	TEXIN2[b]	TEXIN2[c]	TEXIN2[d]
Slope	0.73 ± 0.04	0.98 ± 0.05	0.87 ± 0.04	0.93 ± 0.05	0.92 ± 0.05
Intercept (ppm)	0.14 ± 0.10	0.15 ± 0.12	0.12 ± 0.11	0.12 ± 0.11	0.15 ± 0.11
R^2	0.380	0.420	0.418	0.407	0.431
Avg. Error (ppm)	−0.36	0.12	−0.11	−0.0008	0.0063
Avg. Sq. Error (ppm^2)	2.13	2.86	2.31	2.72	2.41
Number of points	539	539	539	539	539
within 2 ppm	460 (85%)	460 (85%)	470 (87%)	463 (86%)	468 (87%)
within 1 ppm	328 (61%)	352 (65%)	342 (63%)	345 (64%)	348 (65%)

[a]CMA Operations & Design—MOBILE3
[b]CMA Planning Procedure—Short Cut Method
[c]CMA Planning Procedure—MOBILE3
[d]CMA Operations & Design—Short Cut Method

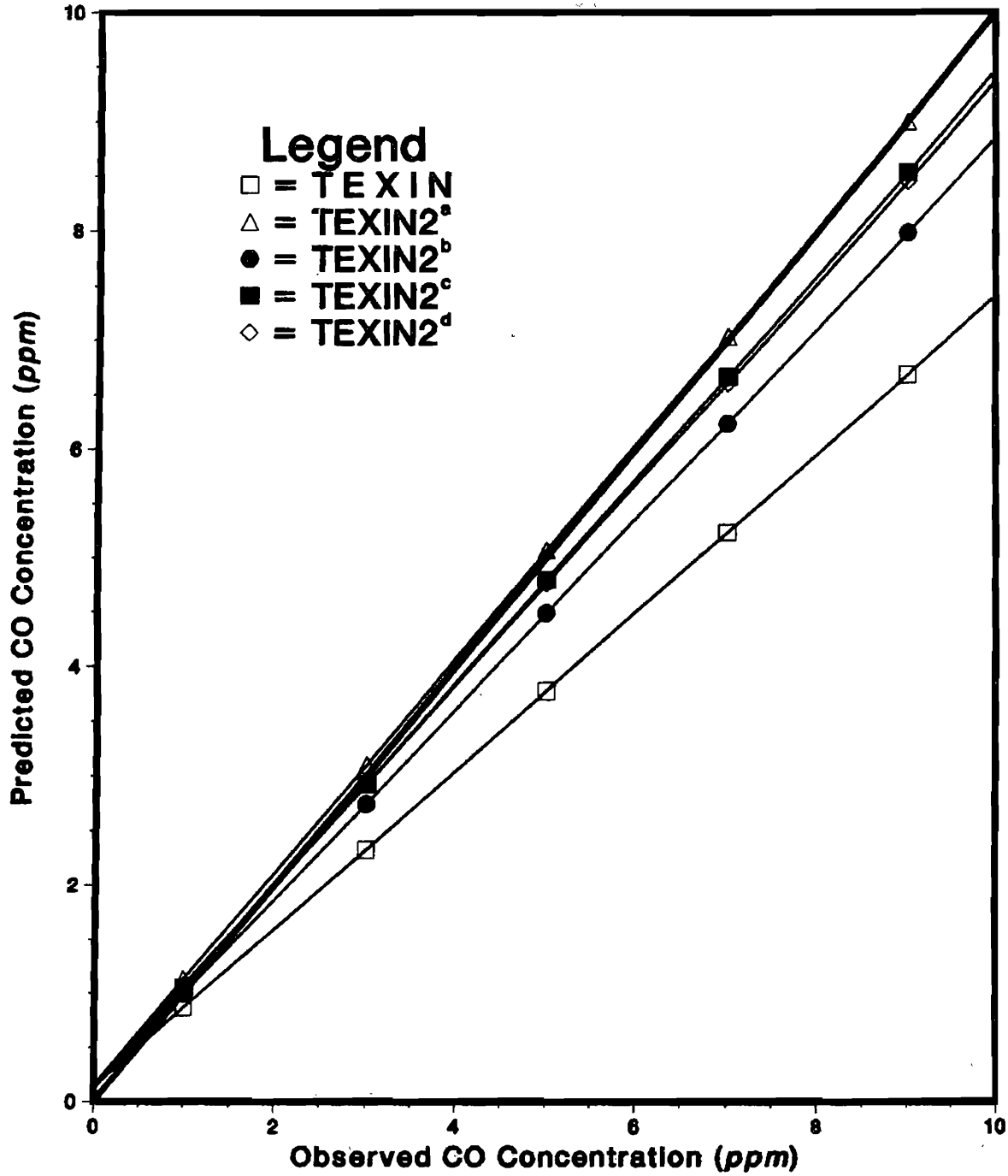


Figure 50

Regression Lines for Various Options of TEXIN2

Using the College Station Data Base

Superscripts Defined in Table 33

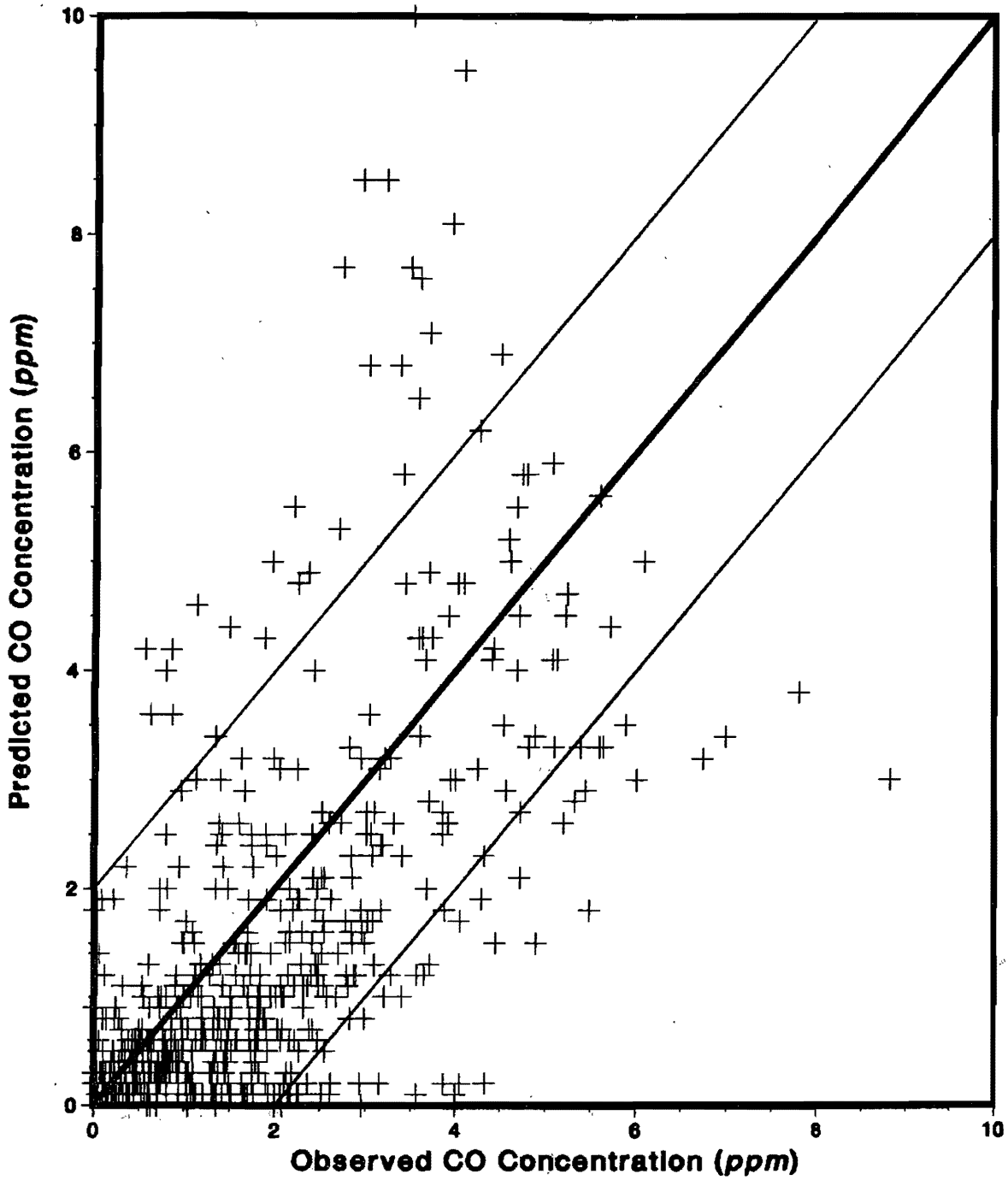


Figure 51

Scatterplot of Original TEXIN Model

Using the College Station Data

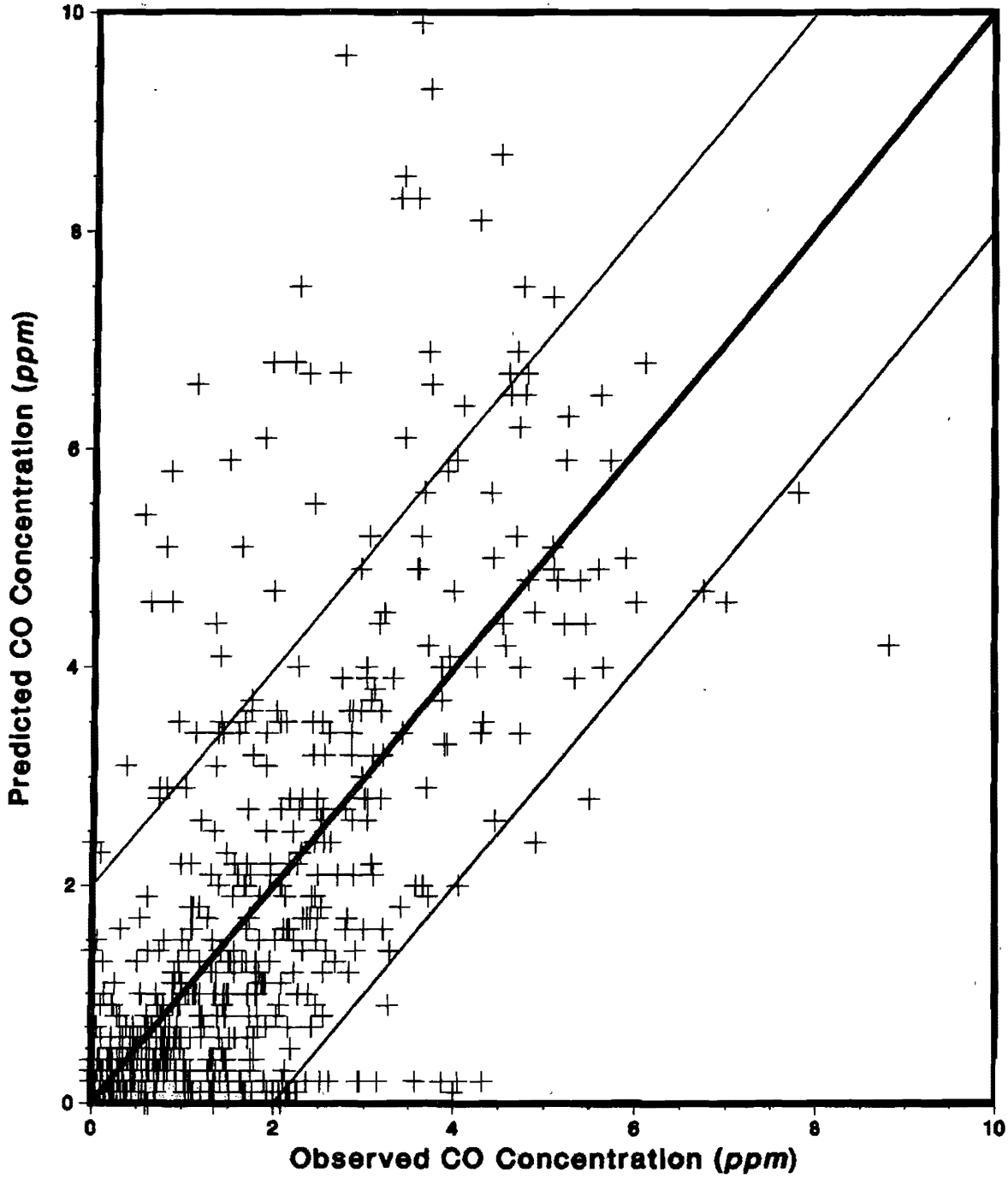


Figure 52

Scatterplot of TEXIN2 Model with CMA Operations and Design and MOBILE3 for the College Station Data

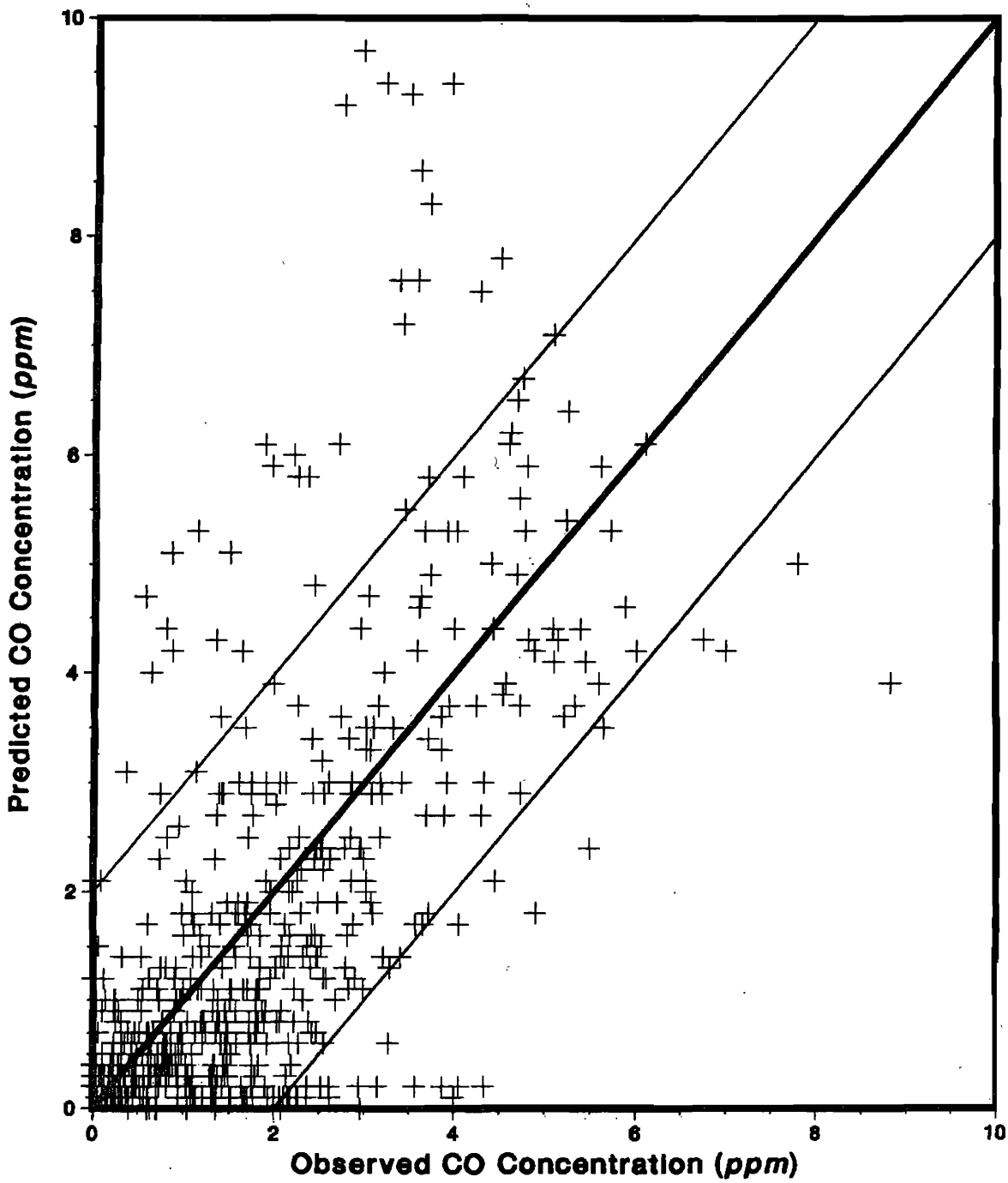


Figure 53

Scatterplot of TEXIN2 Model with CMA Planning

and the Short Cut Method for the College Station Data

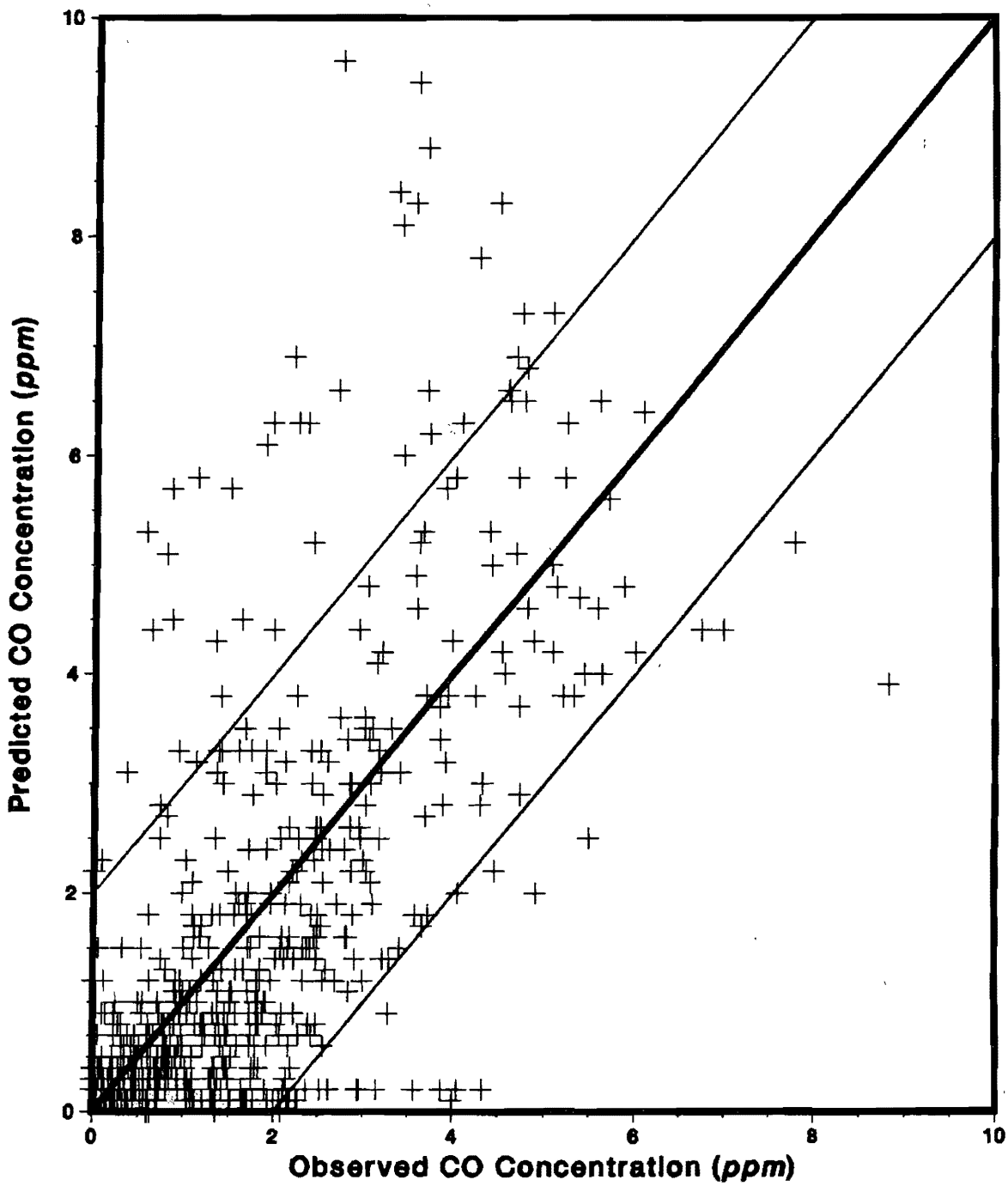


Figure 54

Scatterplot of TEXIN2 Model with CMA Planning and MOBILE3 for the College Station Data

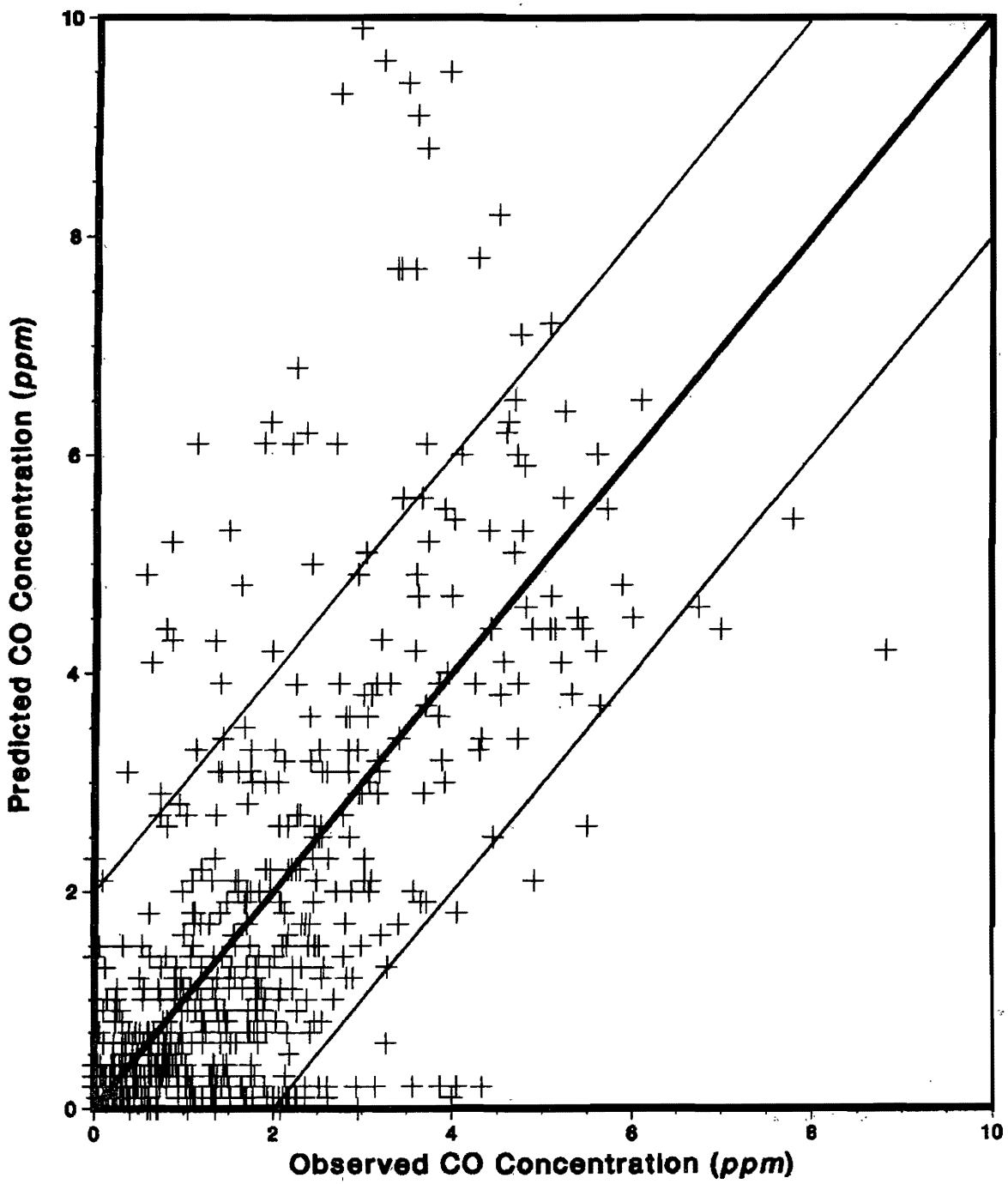


Figure 55

Scatterplot of TEXIN2 Model with CMA Operations and Design and the Short Cut Method for the College Station Data

Appendix F for both the original TEXIN model and the various options used TEXIN2. These scatterplots indicate that while the model error is not very sensitive to wind angle, the error is quite sensitive to wind speed. Scatter where the wind speed is less than 2 m/sec is fairly high while scatter for higher wind speeds is much lower. Also included in Appendix F are scatterplots for the TEXIN2 model for each receptor in the College Station data. These plots indicate that the model may tend to slightly overpredict at lower receptor heights. Furthermore, data scatter appears to be greater at the lower receptors. It is quite evident, however, that the model underpredicts at the higher receptors.

B. Comparison to the California Data Base

The TEXIN and TEXIN2 models were used to simulate the pollutant concentrations in the California data base.[39] All combinations of traffic algorithms and emissions models were used in the comparisons. The data required by the model are described below.

Wind speed, wind angle, and temperature were taken as the appropriate means of the supplied data. The calculated Richardson number, Ri, was used in the determination of atmospheric stability. The stability classes were defined as unstable (Pasquill Class A–B), slightly stable (Pasquill Class C–D), or stable (Pasquill Class E–F) for $Ri < 0$, $0.0 \leq Ri \leq 0.08$, and $Ri > 0.08$, respectively. A surface roughness of 100 cm was estimated from the CALTRANS site description and Table 29. A 1000 m mixing height was used for all simulations.

Information required by TEXIN2 to estimate the traffic parameters is presented in Table 34. All links had exclusive left-turn phases. The widths of all lanes on each approach link were 12 ft. Since vehicle speeds were not determined at the site, a value of 30 mph was used in all cases as estimated by CALTRANS. The fractions of left and right turning vehicles and the local VMT mix were estimated from photography data and are also presented in Table 34. MOBILE3 default values of the percent hot/cold vehicle starts were used in the California data.

The observed carbon monoxide concentration values were calculated as the measured downwind CO concentration minus the average upwind CO concentration. Since the Florin-Freeport intersection site had carbon monoxide monitors in three quadrants, a background concentration could be obtained for winds blowing from any quadrant except for the one without a CO monitor (northeast quadrant). A large majority of the data had winds blowing from one of these three quadrants.

A statistical review of the TEXIN models is given in Table 35 and Figure 56 for the various options invoked in the analysis of the California data. Even though the slope of the regression lines is not as high as the slope from the College Station data, the results are still quite impressive. The average error and scatter in the predictions are very low for all cases presented of the TEXIN2 model. The regression coefficients indicate that over 60% of the experimental data is being explained

Table 34

Input Data for the California Statistical Analyses[65]

Intersection Link Descriptions

Link	Length (m)	Width (m)	Height (m)	Number of Exclusive Lanes		
				Approach	Left Turn	Right Turn
North Freeport	450	23.8	0	2	1	0
South Freeport	450	21.9	0	2	1	0
East Florin	450	23.0	0	2	1	1
West Florin	450	22.6	0	2	1	1

Miscellaneous Data

Parameter	Value
Surface Roughness	100 cm
Averaging Time	60 min
Mixing Height	1000 m
Cycle Length	158 sec
Signal Phases	8
Number of Receptors	16

Table 34 (Continued)

Input Data for the California Statistical Analyses[65]

VMT Mix Data Used in the California Simulations

LDGV[a]	LDGT1[b]	LDGT2[c]	HDGV[d]	LDDV[e]	LDDT[f]	HDDV[g]	MC[h]
.786	.174	.027	.004	.000	.000	.004	.005

[a]Light Duty Gas Vehicles
[b]Light Duty Gas Trucks (GVWR < 6001 lbs)
[c]Light Duty Gas Trucks (GVWR < 8501 lbs)
[d]Heavy Duty Gas Vehicles
[e]Light Duty Diesel Vehicles
[f]Light Duty Diesel Trucks
[g]Heavy Duty Diesel Vehicles
[h]Motorcycles

Turning Fractions for the California Data

	Fraction of Approach Traffic Making Turn from Link							
	West Florin		East Florin		South Freeport		North Freeport	
Time	LT	RT	LT	RT	LT	RT	LT	RT
0600-0700	.170	.060	.000	.291	.233	.033	.417	.208
0700-0800	.142	.057	.014	.261	.122	.034	.352	.154
0800-0900	.117	.051	.019	.240	.122	.067	.426	.183
0900-1000	.119	.056	.044	.285	.307	.071	.525	.275
1000-1100	.123	.086	.061	.254	.255	.123	.415	.289
1100-1200	.141	.067	.040	.291	.277	.132	.495	.256
1200-1300	.116	.064	.028	.238	.258	.152	.481	.260
1300-1400	.097	.060	.038	.272	.303	.079	.505	.284
1400-1500	.121	.063	.027	.252	.281	.083	.485	.273
1500-1600	.095	.071	.041	.282	.298	.099	.476	.308
1600-1700	.055	.057	.024	.223	.247	.137	.401	.295
All Others	.113	.062	.030	.259	.246	.096	.448	.257

by the model.

Scatterplots of the various options used in the TEXIN2 model along with the original TEXIN model are presented in Figures 57–61 for the California data. The scattergrams indicate that the original TEXIN model tends to underpredict for this data base while virtually all data points are within 2 ppm of the experimental data for the revised TEXIN2 model. Scatter is quite low in this data base as is indicated by the magnitude of the average squared error in Table 35. This can also be visualized from the scatterplots by considering the large number of data points on the plot. There are over 9900 points on each of these plots and it would not be too difficult for one to count the number of data points lying outside the 2 ppm lines for the TEXIN2 model.

Appendix F contains scatterplots for different combinations of wind speed and angles for the California data as with the College Station data. Conclusions similar to those from the College Station data can be drawn from the California data with these scatterplots. The scatter present in the model predictions is much greater in low wind speed cases than in high wind speed cases. The general trend for the model to slightly underpredict the California data is also evident from the scatterplots in Appendix F.

C. Comparison to the Houston Data Base

The same options utilized in the verification of the TEXIN2 model with the California data base were used with the Houston data base. The format of the data acquired at the Houston intersection site was quite similar to the data collected at the College Station site. Therefore, the methods used to determine the input parameters for the TEXIN2 model were identical.

Stability class was again determined from insolation data and Figure 48. The mixing height was chosen as 1000 meters. The site was surrounded by tall buildings making it a semi *street canyon* scenario and estimates of the surface roughness difficult. However, an estimate of 4 meters (400 cm) was chosen in the analyses. Table 36 gives a physical description of the intersection that was used as input to the TEXIN2 model. Only the Woodway Boulevard legs had exclusive left-turn phases. Two additional non-delayed links were used by the model to approximate the curves in the eastern leg of Woodway while one additional non-delayed link was used for the western Woodway leg. The length of each Woodway leg shown in Table 36 is the length measured along the roadway. All lanes on each link were 12 ft in width. The approach traffic volumes, turning fractions, and average vehicle speeds were obtained from traffic loop counters installed at the Houston site. The input for VMT mix and the percent hot/cold starts were county-wide values obtained from the Texas State Department of Highways and Public Transportation and are tabulated in Table 37.

The observed carbon monoxide concentrations were calculated as the measured downwind CO concentration minus the average upwind CO concentration. As at the College Station site, one tower (Tower 1) was set up as the primary upwind tower. Tower 1 was in the southeast quadrant

Table 35

TEXIN and TEXIN2 Model Results

A Comparison of the Various Options Available in TEXIN2

California Data Base

Statistic	TEXIN	TEXIN2[a]	TEXIN2[b]	TEXIN2[c]	TEXIN2[d]
Slope	0.49 ± 0.007	0.65 ± 0.006	0.70 ± 0.006	0.64 ± 0.006	0.70 ± 0.006
Intercept (ppm)	0.08 ± 0.008	0.12 ± 0.008	0.12 ± 0.008	0.12 ± 0.008	0.13 ± 0.008
R^2	0.628	0.613	0.616	0.612	0.618
Avg. Error (ppm)	−0.21	−0.08	−0.05	−0.09	−0.05
Avg. Sq. Error (ppm^2)	0.54	0.48	0.49	0.49	0.49
Number of points	9930	9930	9930	9930	9930
within 2 ppm	9611 (97%)	9662 (97%)	9662 (97%)	9662 (97%)	9665 (97%)
within 1 ppm	8968 (90%)	8998 (91%)	8992 (91%)	8993 (91%)	8992 (91%)

[a]CMA Operations & Design—MOBILE3
[b]CMA Planning Procedure—Short Cut Method
[c]CMA Planning Procedure—MOBILE3
[d]CMA Operations & Design—Short Cut Method

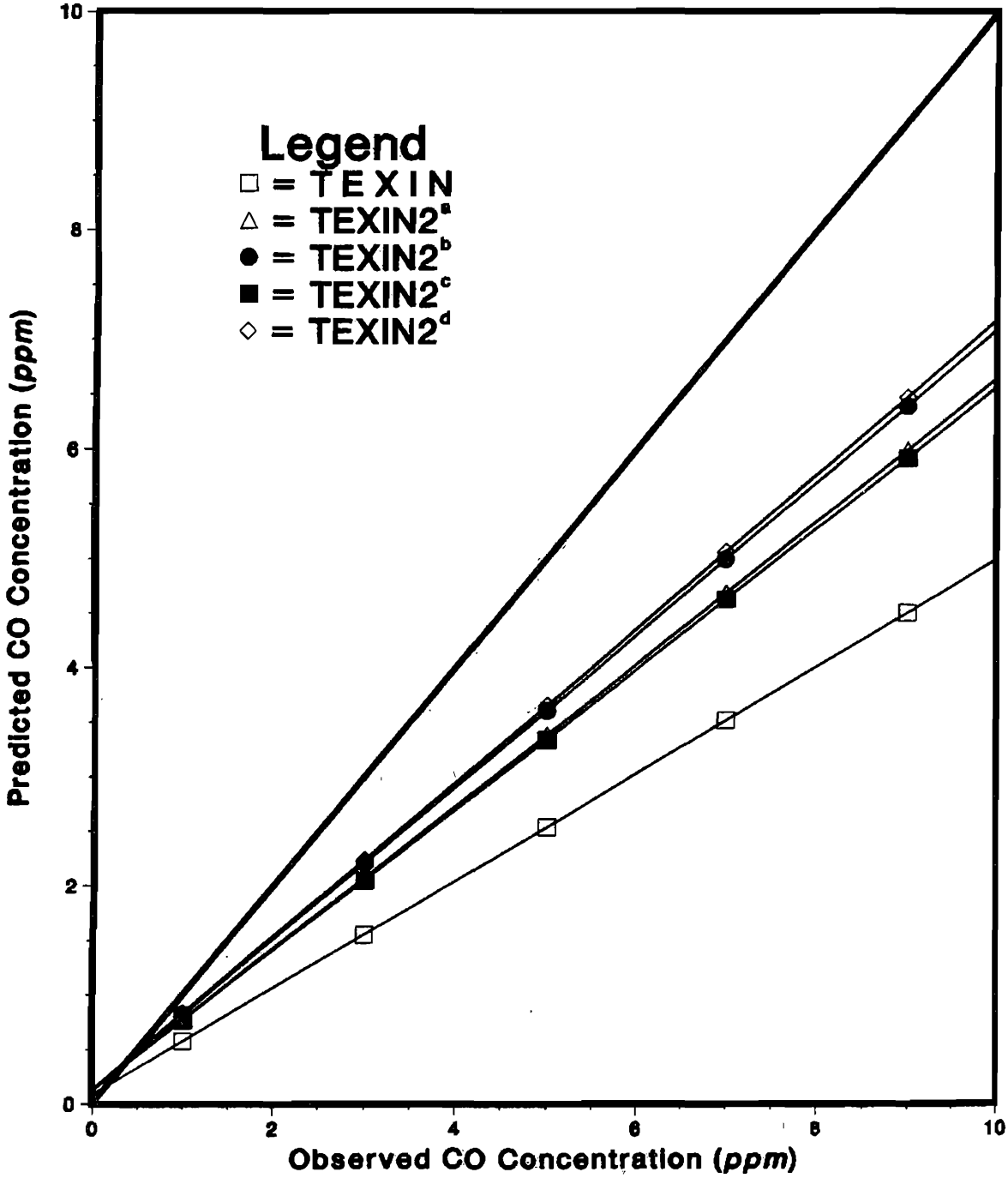


Figure 56

Regression Lines for Various Options of TEXIN2

Using the California Data Base

Superscripts Defined in Table 35

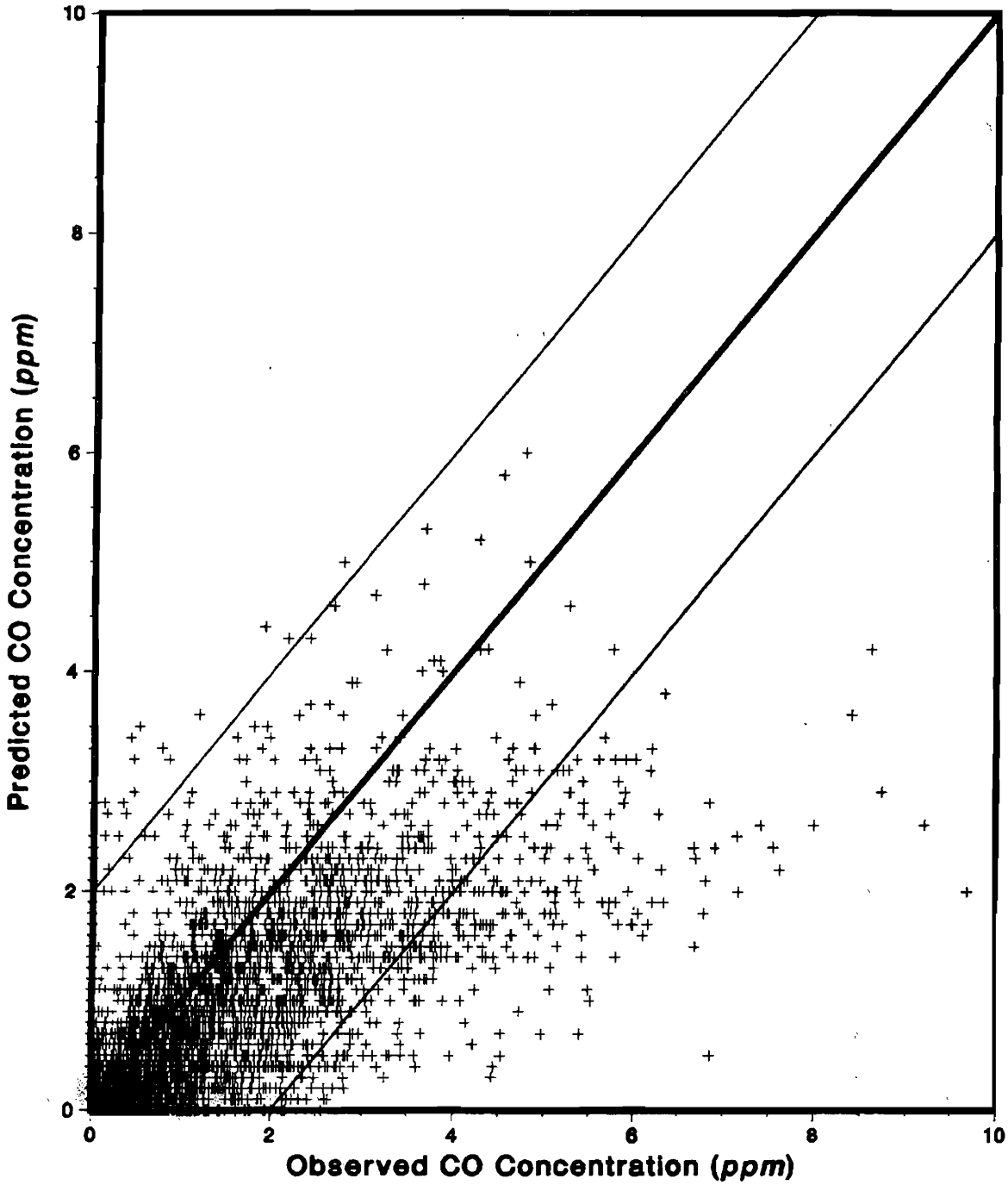


Figure 57

Scatterplot of Original TEXIN Model

Using the California Data

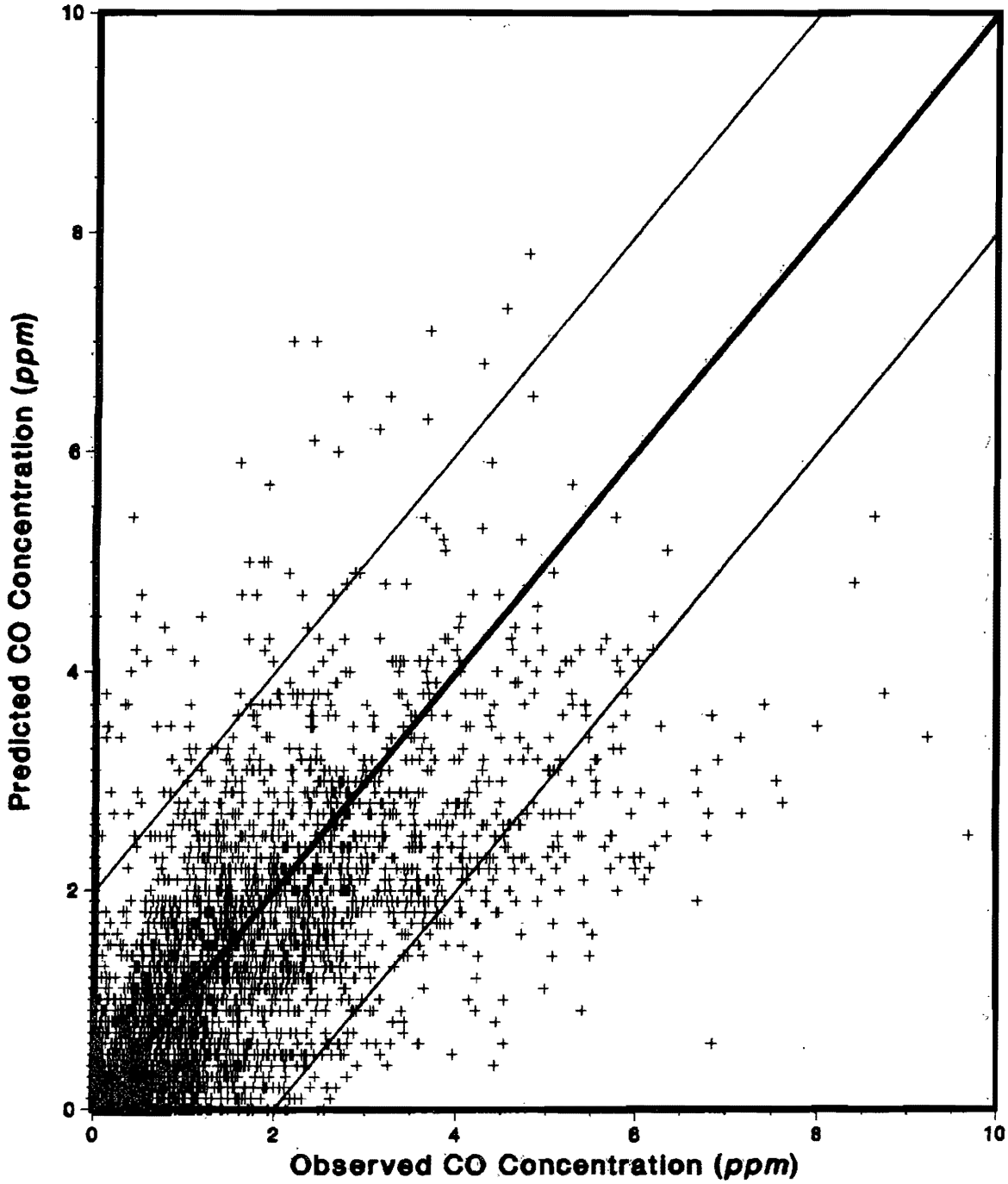


Figure 58

Scatterplot of TEXIN2 Model with CMA Operations and Design and MOBILE3 for the California Data

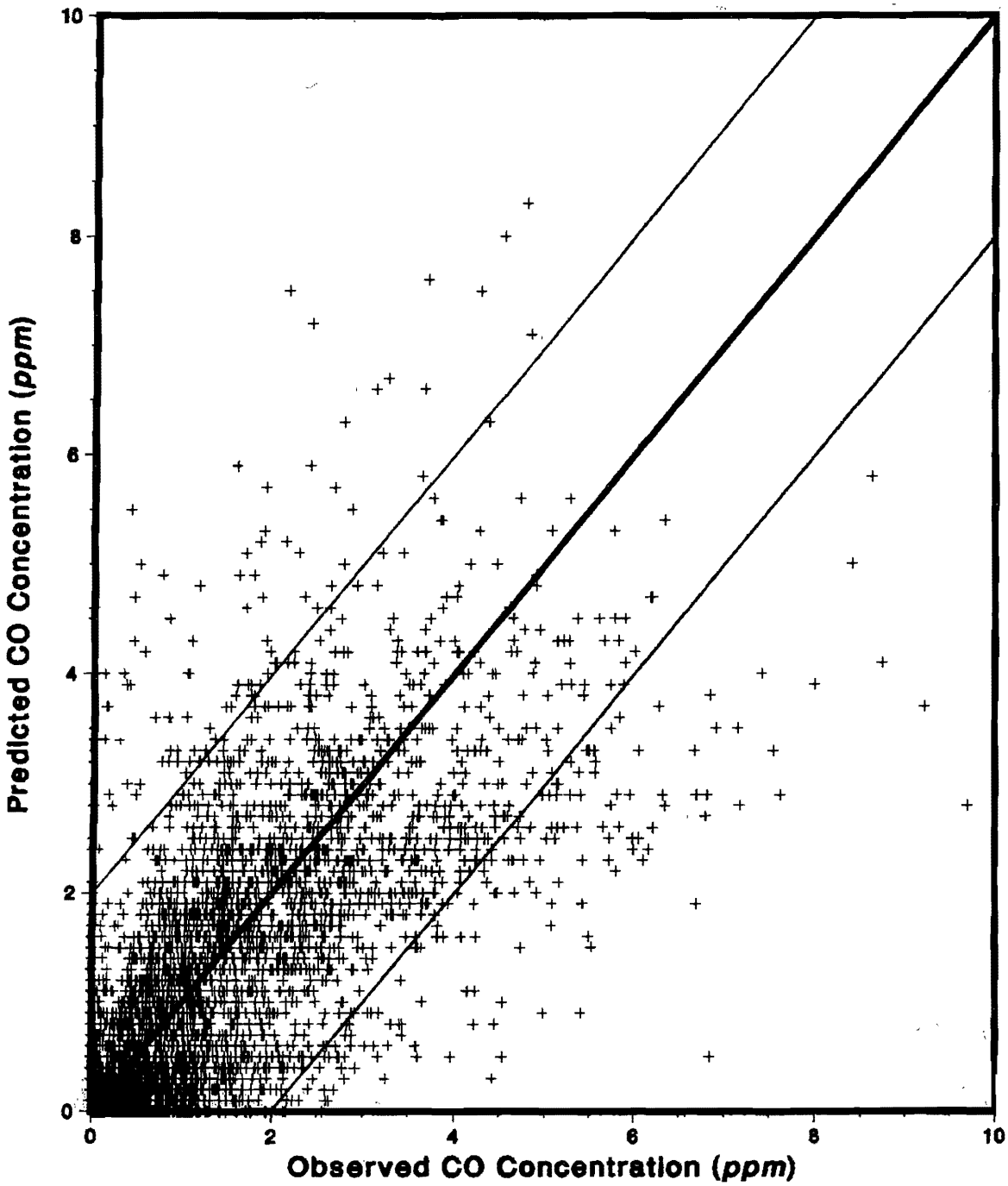


Figure 59

Scatterplot of TEXIN2 Model with CMA Planning and the Short Cut Method for the California Data

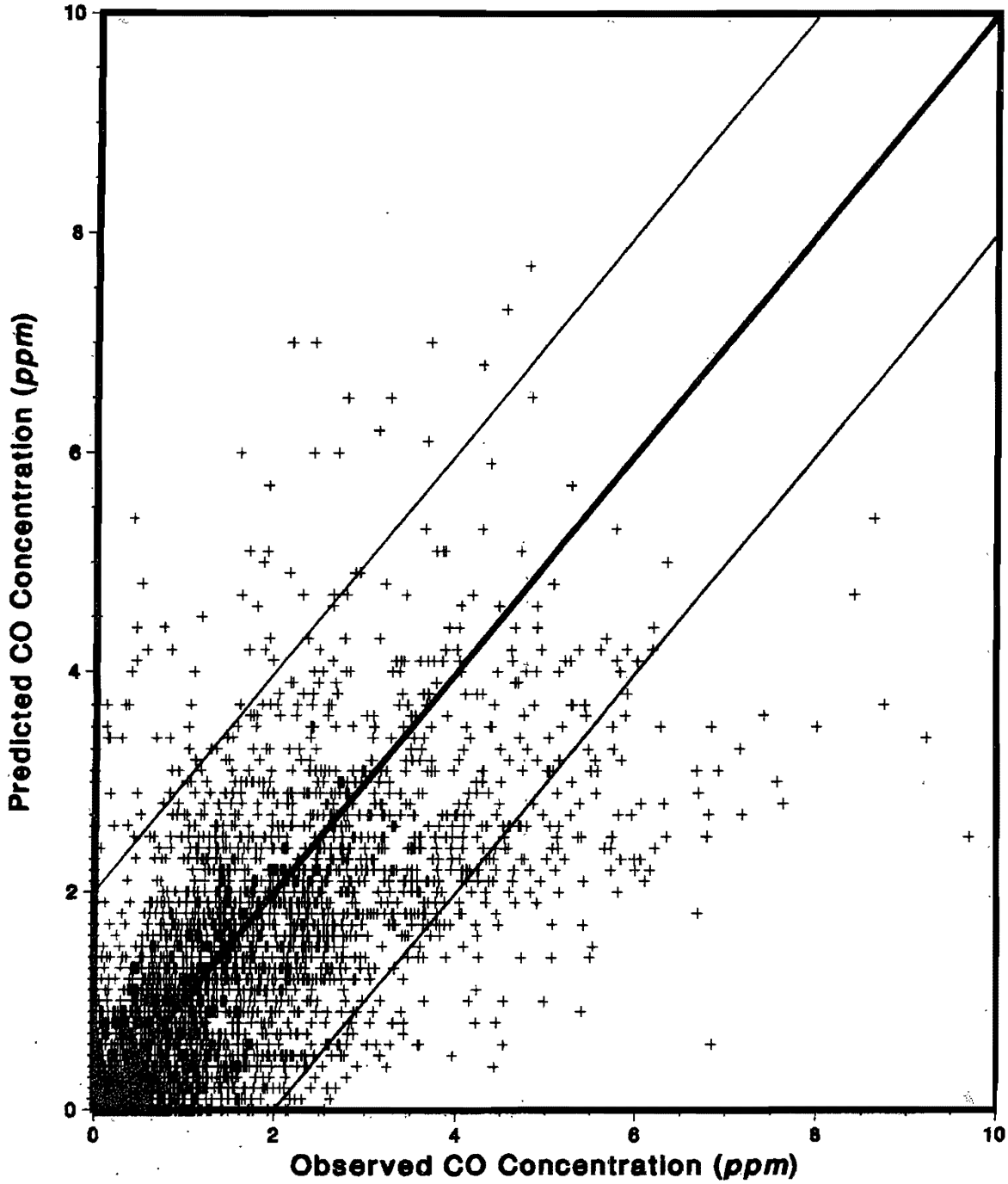


Figure 60

Scatterplot of TEXIN2 Model with CMA Planning

and MOBILE3 for the California Data

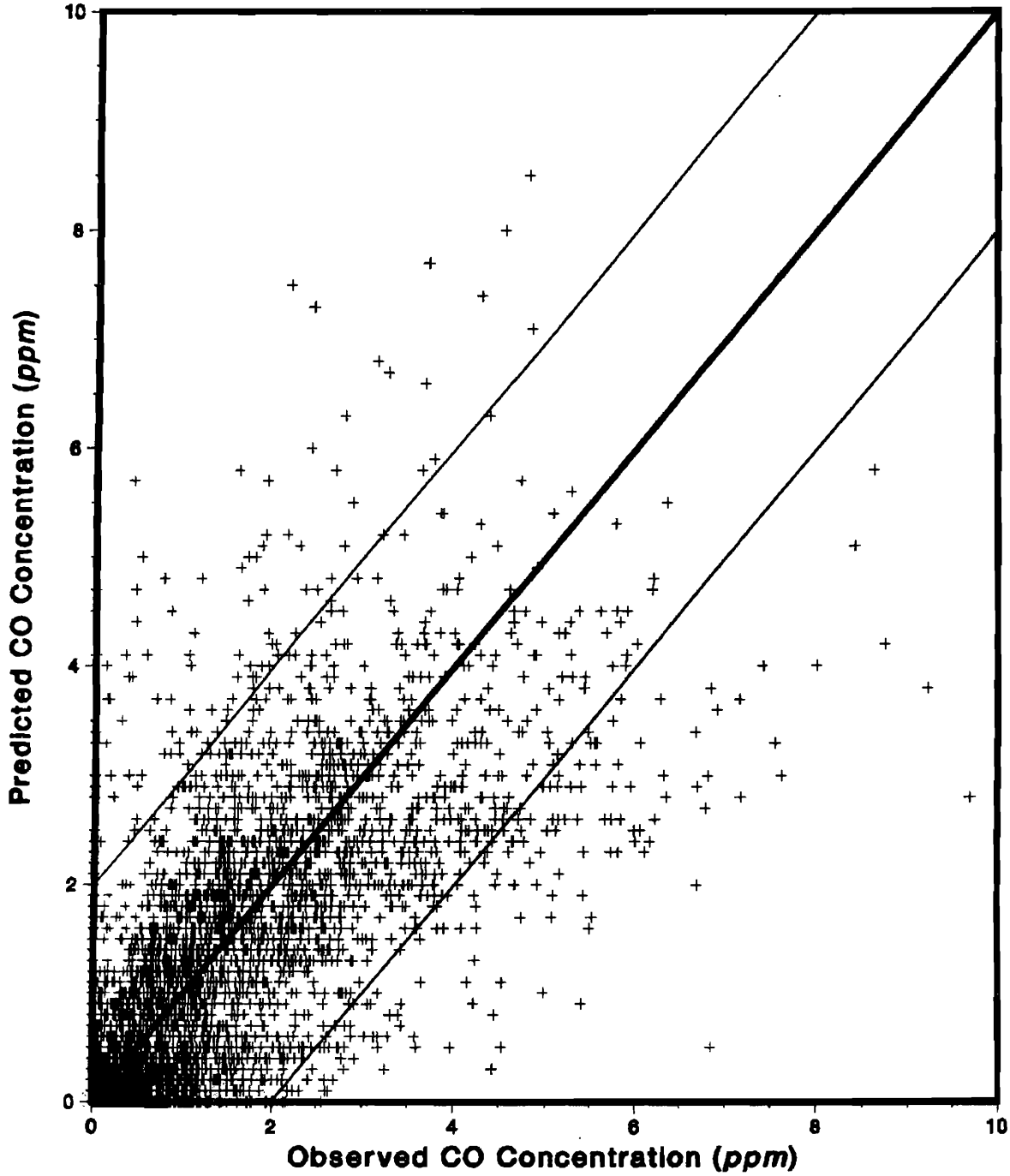


Figure 61

Scatterplot of TEXIN2 Model with CMA Operations and Design and the Short Cut Method for the California Data

Table 36

Input Data for the Houston Statistical Analyses

Intersection Link Descriptions

Link	Length (m)	Width (m)	Height (m)	Number of Exclusive Lanes		
				Approach	Left Turn	Right Turn
North S. Post Oak	300	7.3	0	1	0	0
South S. Post Oak	750	12.5	0	2	0	0
East Woodway	688	18.6	0	2	1	0
West Woodway	519	19.2	0	2	1	1

Miscellaneous Data

Parameter	Value
Surface Roughness	400 cm
Averaging Time	60 min
Mixing Height	1000 m
Cycle Length	80 sec
Signal Phases	5
Number of Receptors	9

Table 37

Motor Vehicle Data Used in the Houston Statistical Analyses

VMT Mix for Harris County, Texas

Year	LDGV[a]	LDGT1[b]	LDGT2[c]	HDGV[d]	LDDV[e]	LDDT[f]	HDDV[g]	MC[h]
1981	.743	.127	.082	.020	.007	.001	.012	.008

[a]Light Duty Gas Vehicles
[b]Light Duty Gas Trucks (GVWR < 6001 lbs)
[c]Light Duty Gas Trucks (GVWR < 8501 lbs)
[d]Heavy Duty Gas Vehicles
[e]Light Duty Diesel Vehicles
[f]Light Duty Diesel Trucks
[g]Heavy Duty Diesel Vehicles
[h]Motorcycles

Percent of Hot/Cold Starts for 1981 in Harris County, Texas

Hour	PCCN[a]	PCHC[b]	PCCC[c]
00–02	22	7	27
03–05	25	7	26
06–08	22	10	24
09–11	10	18	18
12–14	9	18	18
15–17	13	14	19
18–20	8	17	20
21–23	12	11	25

[a]Percent VMT accumulated by cold start non-catalyst vehicles
[b]Percent VMT accumulated by hot start catalyst vehicles
[c]Percent VMT accumulated by cold start catalyst vehicles

Source: Texas State Department of Highways and Public Transportation

so that for only those cases with a wind out of the southeast quadrant (winds blowing from any angle between the eastern leg of Woodway Boulevard and the southern leg of South Post Oak Lane in Figure 22) could a true background concentration be properly determined. Since Tower 1 was relatively close to South Post Oak Lane, only those cases with a wind from the southeast quadrant were used in the analyses. Approximately two-thirds of the data had winds blowing from this quadrant.

The TEXIN models were statistically compared for 97, 60-minute cases of the Houston data base. The results of these comparisons are presented in Table 38 and Figure 62. The statistics for the Houston data base are the worst of the three data bases. The TEXIN2 model clearly overpredicts and exhibits a large amount of scatter for each case presented. The predictions made by the original TEXIN model are better than those of the revised model.

Scatterplots for the TEXIN2 options given in Table 38 as well as for the original TEXIN model are illustrated in Figures 63–67. The scatterplots do indeed demonstrate that the model overpredicts the experimental data and exhibits a large amount of data scatter.

The reduced performance of the model using the Houston data may be attributed to the geometry of the site. The TEXIN model was not developed for street canyon scenarios. Attempts to improve the performance and versatility of the original model focused on reduction in scatter and improved predictive capabilities for similar scenarios. The chaotic air movement present in street canyon intersections cannot be adequately described by any intersection models investigated in this study. The final result is that of improved accuracy of TEXIN2 for at-grade sites such as College Station and Sacramento, and much more scatter at complex street canyon sites characterized by the Houston data base.

Table 38

TEXIN and TEXIN2 Model Results

A Comparison of the Various Options Available in TEXIN2

Houston Data Base

Statistic	TEXIN	TEXIN2[a]	TEXIN2[b]	TEXIN2[c]	TEXIN2[d]
Slope	0.79 ± 0.05	1.15 ± 0.09	0.98 ± 0.08	1.14 ± 0.09	0.99 ± 0.08
Intercept (ppm)	1.04 ± 0.18	1.35 ± 0.29	1.55 ± 0.28	1.34 ± 0.29	1.56 ± 0.28
R^2	0.468	0.464	0.400	0.464	0.399
Avg. Error (ppm)	0.48	1.73	1.50	1.71	1.53
Avg. Sq. Error (ppm^2)	3.54	9.80	8.64	9.68	8.79
Number of points	295	295	295	295	295
within 2 ppm	235 (80%)	174 (59%)	183 (62%)	176 (60%)	184 (62%)
within 1 ppm	156 (53%)	116 (39%)	117 (40%)	115 (39%)	117 (40%)

[a]CMA Operations & Design—MOBILE3
[b]CMA Planning Procedure—Short Cut Method
[c]CMA Planning Procedure—MOBILE3
[d]CMA Operations & Design—Short Cut Method

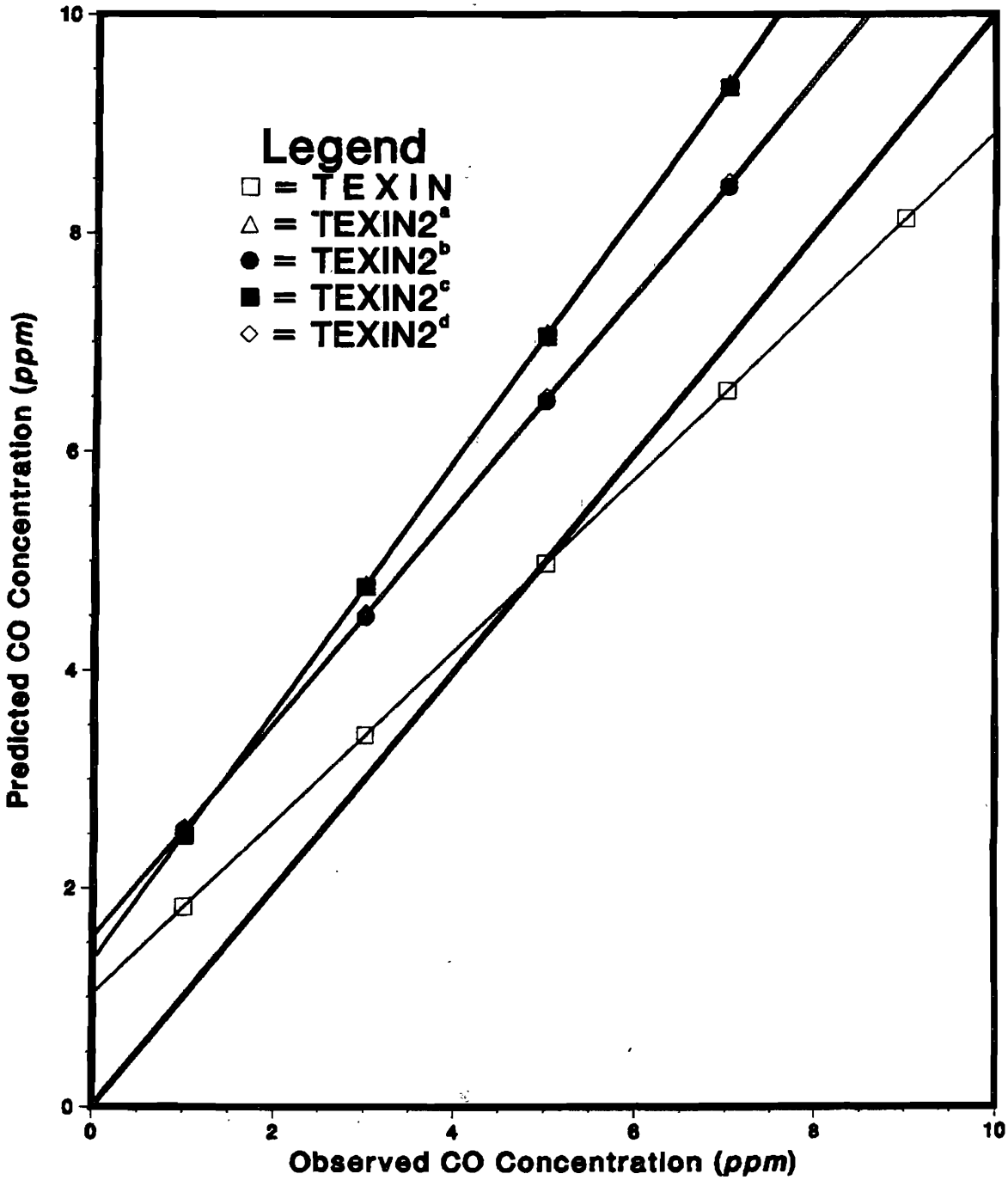


Figure 62

Regression Lines for Various Options of TEXIN2

Using the Houston Data Base

Superscripts Defined in Table 38

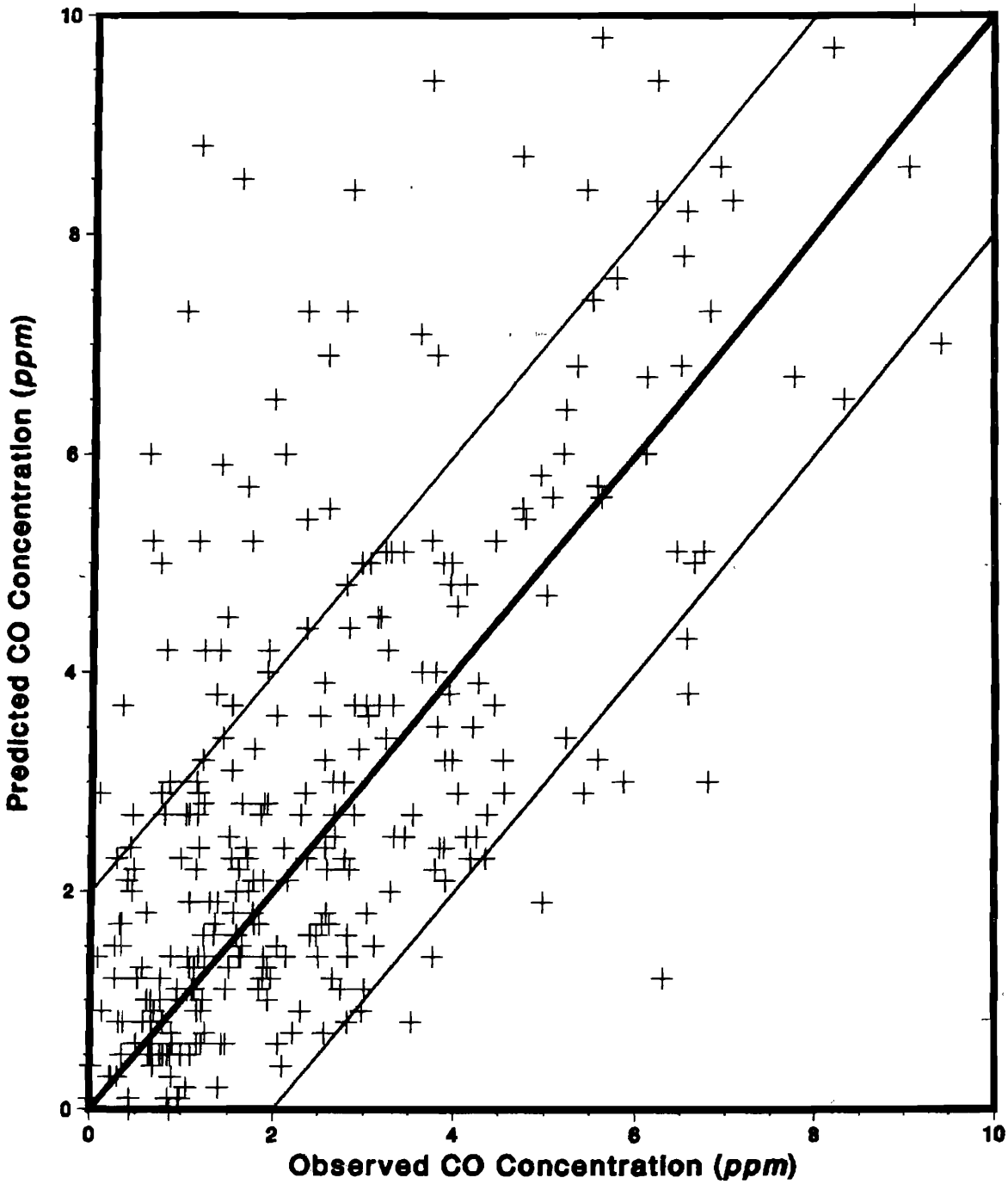


Figure 63

Scatterplot of Original TEXIN Model

Using the Houston Data

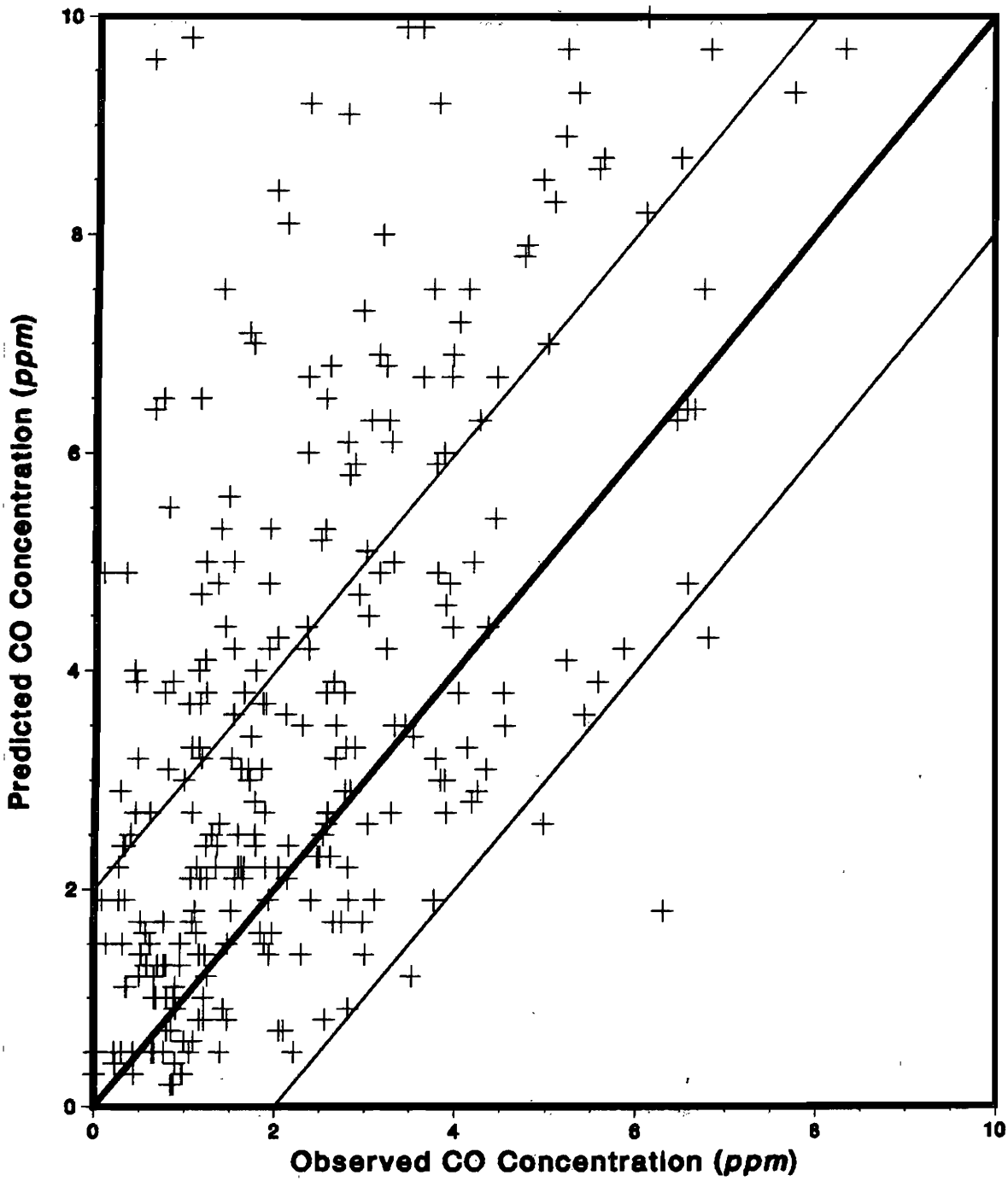


Figure 64

Scatterplot of TEXIN2 Model with CMA Operations and Design and MOBILE3 for the Houston Data

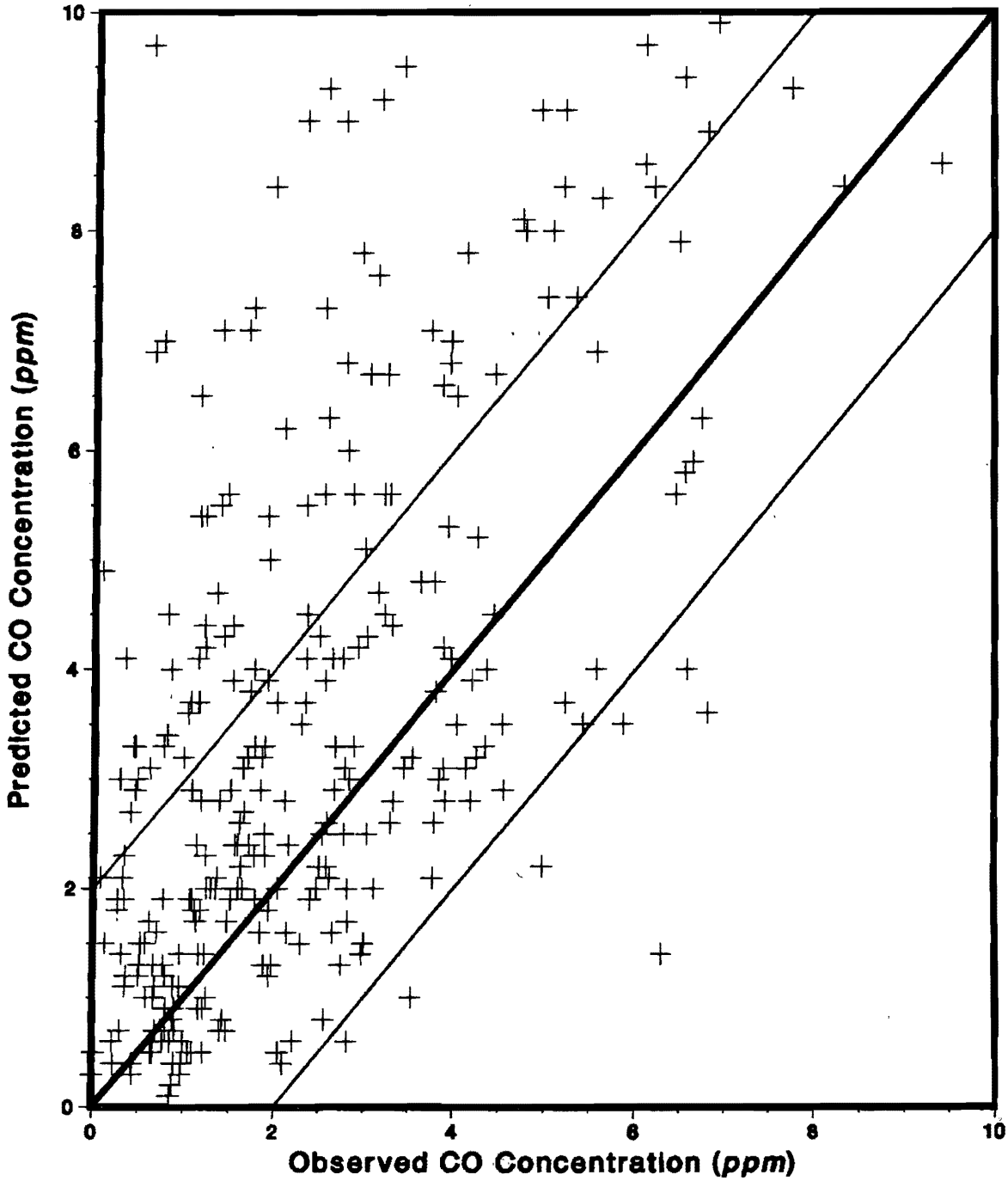


Figure 65

Scatterplot of TEXIN2 Model with CMA Planning and the Short Cut Method for the Houston Data

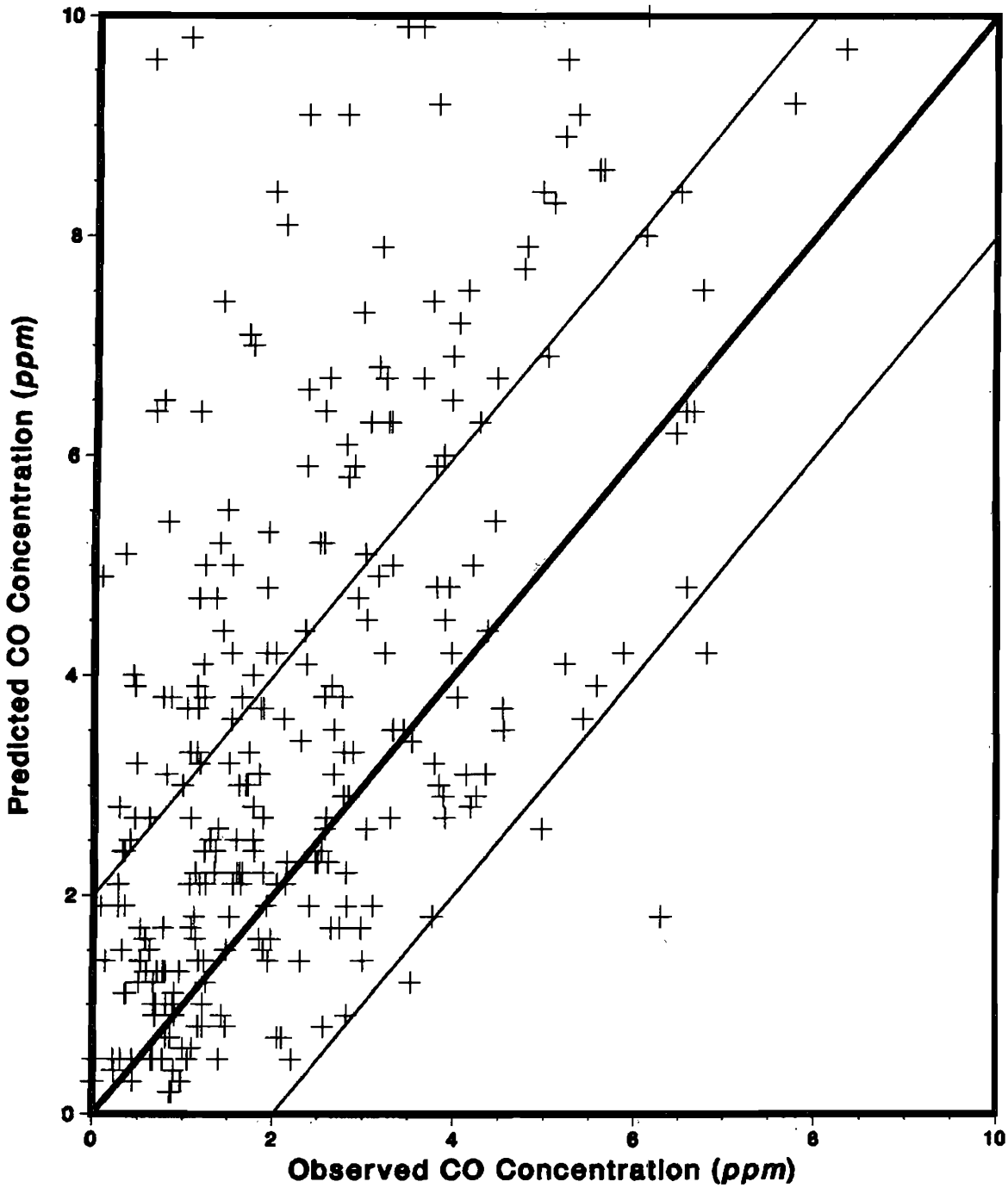


Figure 66

Scatterplot of TEXIN2 Model with CMA Planning

and MOBILE3 for the Houston Data

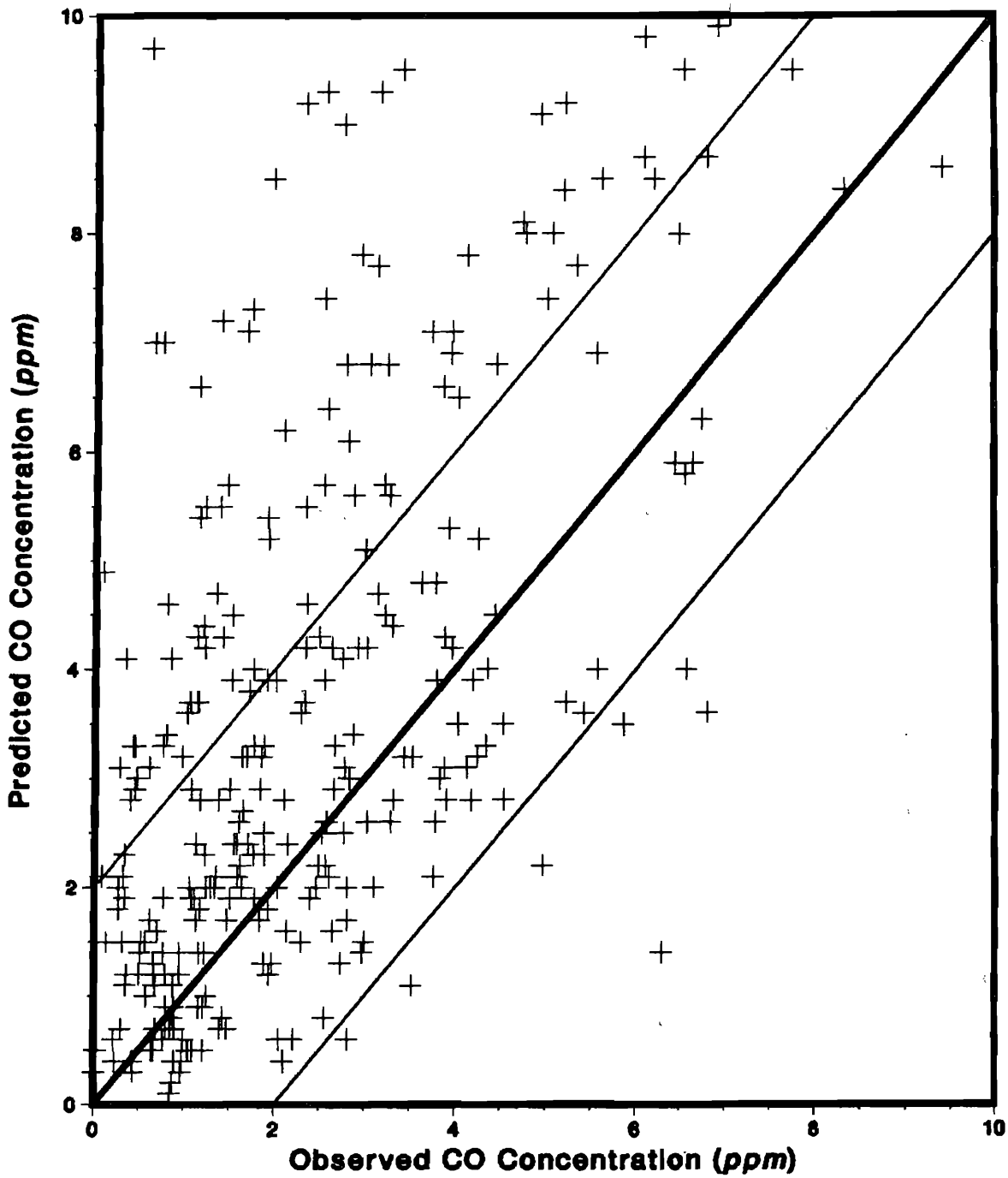


Figure 67

Scatterplot of TEXIN2 Model with CMA Operations and Design and the Short Cut Method for the Houston Data

III. Experimental Results

Three major tasks were performed during the experimental phase of the project. The first of these involved the establishment of a large data base that can be used for model verification and development. Secondly, the mass balance technique was applied to the data to calculate vehicular emission factors. Finally, the data from the tracer gas experiments were used to investigate the dispersion process along roadways and to compare actual tracer emission rates to rates calculated from the mass balance technique. These topics are discussed below after a summary of instrument accuracy.

A. Analysis of Data Accuracy

In any data collection process, there are many sources of error. Every instrument has errors associated with it and, in addition, the data acquisition system has errors associated with it. Furthermore, systems that require large quantities of electrical cable will have inherent noise and line losses. Table 39 lists the errors associated with each of the instruments. This section is concerned with the methods of establishing those error limits.

Analog to Digital Converter

The data acquisition system for this project employed two 12 bit analog to digital converters (A/D) in the Balcones computer. Errors in this unit may arise from two possible sources. First the span could drift, causing information read on the circuit board to be some factor greater or less than the actual voltage. This drift reaches its maximum value at the maximum input to the A/D and vanishes at a data reading of zero. The second type of error associated with the A/D involves the reading of an apparent voltage when the input was zero. The data would then be offset by a fixed constant.

In this project, the A/D calibration was checked frequently using a simple calibration program that allowed for monitoring the A/D integer counts while a known voltage was supplied to the converter. The span or zero drift on the circuit board rarely exceeded 0.3%. This value was termed insignificant when compared to other errors.

Meteorological Instruments

Horizontal Anemometers

There were two primary sources of error associated with this instrument. The starting threshold for these instruments was relatively high (0.75 mph). This meant that under low wind speed, the recorded speed was lower than the actual wind speed. The second source of error was due to the mass of the anemometer cups. During gusty conditions, the inertia of the cups would cause the anemometer to continue to rotate even after the gust had subsided. Therefore, under these conditions, the recorded wind speed was greater than the actual wind speed. The manufacturer reported an accuracy of $\pm 2\%$ of full scale for these instruments. Since the anemometers employed

Table 39

An Analysis of Instrument Accuracy

Instrument	Error
A/D	0.3% of span or drift
Radars	
Overall Count	4%
Speed	10% of reading
UVW Anemometers	
Vertical	3%
Horizontal	1%
Cup Anemometers	2%
Wind Vanes	5°
Thermometers	1.5°
Psychrometer	3% Relative Humidity
Pyranometer	10%
Barometer	0.02" Hg
Ecolyzers	0.5 ppm
Gas Chromatographs	
CO_2	10 ppm
All others	0.1 ppm
Ozone	0.001 ppm
DASIBI CO	0.1 ppm
NO_x	0.002 ppm

a photo-chopper technique, the electronic calibration of these instruments could be easily checked with a signal generator and an oscilloscope. Rarely did the electronic drift of the anemometer exceed ±0.1%.

Wind Vanes

The primary error for these instruments was due to the alignment procedure used during installation. The vanes were aligned by choosing landmarks and directing the vane towards the object. This procedure was probably accurate to within 5°. Since the standard deviation of the wind angle was usually high, this alignment error was considered negligible. A second source of error on the wind vanes was due to the dead band that was present in the sensing potentiometer. The manufacturer reported this dead band as 3°. Since the full scale range of the wind vanes employed was 360°, this error occurred during prevailing northerly winds.

UVW and Vertical Anemometers

The error for these instruments was not checked by project personnel. The manual supplied with the instruments stated the accuracy was ±1% for a horizontal position and ±3% for a vertical position. The main inaccuracy with these instruments was due to the fact that the propellers did not strictly have a true cosine response to the wind angle. This deviation was the greatest at wind angles of 45° with respect to a particular component. In order to combat this non-cosine response, the data were corrected by applying response factors. These factors were reported by Gill[66] and have been investigated by many others. The corrections involved a comparison of the ratio of the magnitude of the horizontal components of the wind speed. The correction factors used in the data reduction program are presented in Table 40. The starting threshold for these anemometers was 0.1–0.2 meter/sec. The UVW anemometers also suffered from the same alignment error as the wind vanes.

Thermometers

The manufacturer claimed that these instruments were accurate to 0.5°F. The electronics calibration of the thermometers was regularly checked. An earlier TTI project indicated that the solar shielding around the thermometers was not adequate to claim a 0.5°F accuracy.[11] They stated that a better accuracy estimation was 1.5°F.

Psychrometers

The project personnel did not check the accuracy of the psychrometers. The manual stated that the instruments were accurate to within 3% relative humidity. The calibration of the psychrometers was occasionally checked using a wet and dry bulb thermometer.

Pyranometers

The manufacturer claimed that the pyranometer is accurate to within ±10% of the industry

Table 40

Non-Cosine Response Factors for the UVW Anemometer[66]

Ratio of Magnitude of Smaller Horizontal Component to Larger Horizontal Component	Multiply Larger Component by	Multiply Smaller Component by
0.00	1.000	1.250
0.05	1.000	1.250
0.10	1.000	1.250
0.15	1.002	1.250
0.20	1.004	1.250
0.25	1.008	1.250
0.30	1.013	1.248
0.35	1.018	1.245
0.40	1.024	1.240
0.45	1.030	1.235
0.50	1.037	1.228
0.55	1.043	1.22
0.60	1.049	1.212
0.65	1.057	1.203
0.70	1.066	1.193
0.75	1.083	1.183
0.80	1.093	1.173
0.85	1.093	1.163
0.90	1.103	1.152
0.95	1.115	1.141
1.00	1.130	1.130

standard 48 junction thermopile black and white pyranometer. Errors arose from a slight non-cosine response to the angle of incidence and blockage of light by taller trees on the south side of the roadway.

Barometer

The barometer was accurate to within ±0.02" Hg over any ±2" Hg span. The barometer was occasionally calibrated using a mercury barometer.

Air Monitoring Instruments

Ecolyzers

Since the carbon monoxide concentrations were the primary goal of this project, it was considered quite important to establish the level of accuracy for these instruments. Previous projects have indicated that the span and zero drifts over long intervals of time were large enough to severely degrade the CO data. Therefore, methods were developed so that spans and zeros from the instruments could be recorded on tape for application of calibration factors to the data in the data reduction phase. These factors assumed linear calibration drifts. The success of this procedure was checked in an earlier study.[11] Two instruments were run side by side for several days during the study. The Ecolyzers were treated no different than any other Ecolyzer on the project. The data reduction program was then used to apply calibration factors to the data. A plot of the results is shown in Figure 68. The Ecolyzers agreed with each other quite well which therefore indicates the success of the application of calibration drift factors. The report further indicated that the average error for the Ecolyzers may be as low as 0.3 ± 0.25 ppm. However, the quoted error of ±0.5 ppm on these instruments is used as the stated error bounds.

Byron Gas Chromatographs

Byron stated that these instruments were accurate to within ±1% of full scale which is translated to ±0.1 ppm for THC, non-methane hydrocarbons, carbon monoxide, and methane and ±10 ppm for carbon dioxide. Problems with the carbon monoxide analytical cycle would point to the fact that the error on CO concentrations was greater than the stated value. Calibration drift factors were also applied to this instrument; however, the drift was small for most gases so that the calibration frequency of the Byron was about every two days.

Ozone Analyzers

The manufacturer claimed that these instruments were accurate to ±0.001 ppm. The major source of error in ozone monitoring was due to decomposition of the sample before it reached the analyzer. Frequent checks were made on the cleanliness of the optical system in the monitor.

DASIBI Carbon Monoxide Analyzers

The stated accuracy of this instrument was ±0.1 ppm. Correction factors from calibration

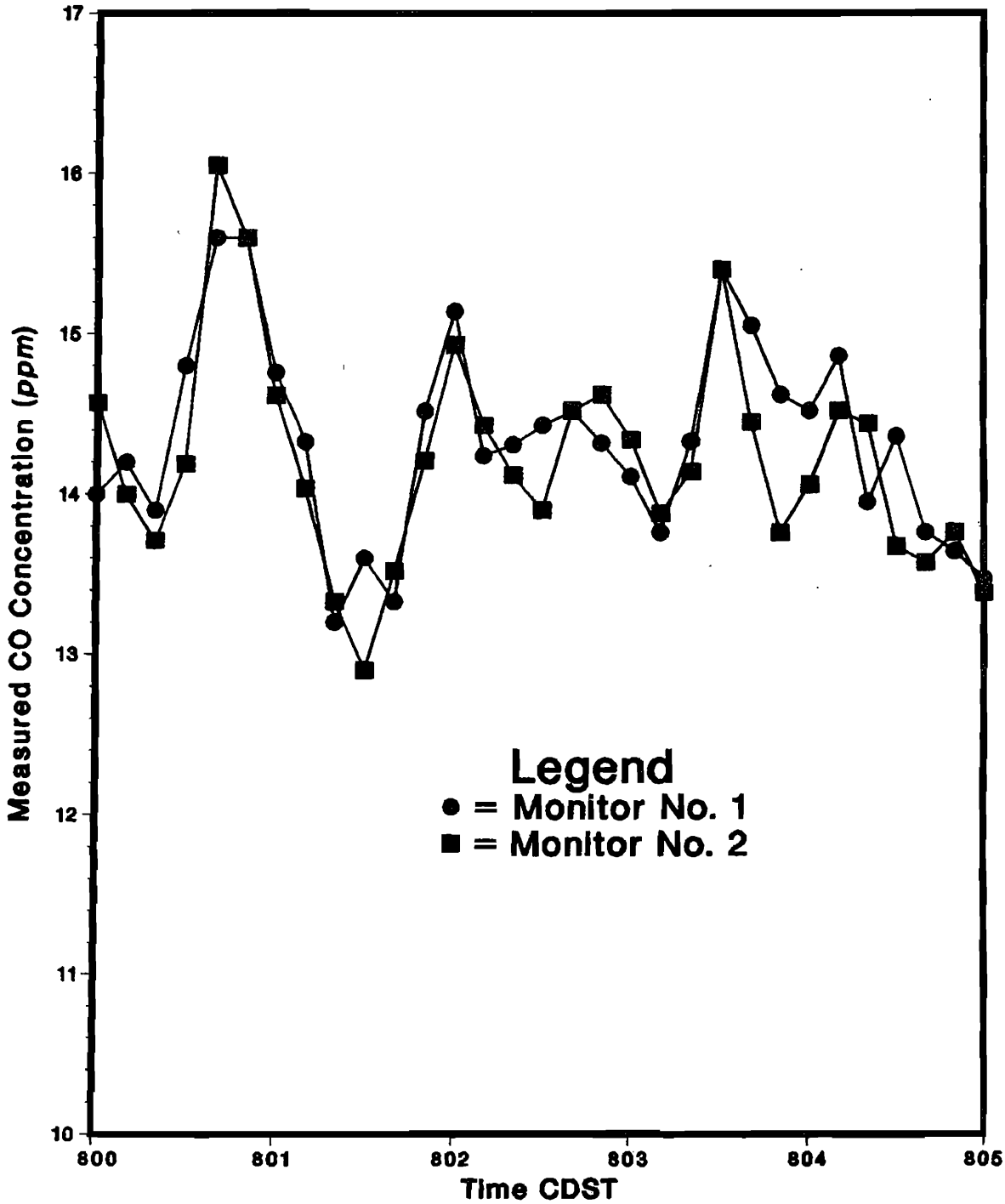


Figure 68

Verification of Calibration Drift Factor Application

data were applied to the DASIBI CO monitor readings as with the Ecolyzers. The drift on this instrument was quite small and reported CO concentrations were highly accurate. Calibration frequency on the monitor was about every two days.

NO_x Monitors

The manufacturer stated that the NO_x analyzers were precise to within ± 0.0015 ppm, with a maximum zero drift of ± 0.002 ppm. The true accuracy of these instruments was also a function of the ability of the calibrator to deliver the proper concentration of diluted NO span gas. The NO_x analyzers were calibrated about every two days.

Tracer Gas Studies

The Hastings mass flowmeters used to determine the SF_6 emission rate were accurate to within $\pm 1\%$ of full scale for air flow measurements. The readings from this instrument were multiplied by 0.28 to account for the difference in heat capacity between air and SF_6.

To insure that the calibration SF_6 gas mixture was suitable, two sources of the gas were used. The calibration mixture used by the electron capture detector (2.02 ppb SF_6) was obtained from Scott-Marrin. This standard was compared to a 2.0 ppb standard produced by Matheson. The gases were checked by Radian Corporation for analysis accuracy. A copy of the Radian report is included in Appendix E.

The emission rates of SF_6 were measured both prior to and after each tracer gas experiment. This step was taken to insure that a constant flow rate of tracer was emitted throughout the course of the experiment. These measurements indicated that inconsistencies in emission rate during the driving period were negligible.

Traffic Monitoring

The errors associated with the traffic monitoring system were due to the fact that more than a simple global traffic count was desired. The radars gave information on the average speed, traffic volume, and vehicle distribution. Since the distribution of vehicle sizes is extremely important in a roadway air quality study, every effort was made to record these data accurately.

There were four possible sources of error associated with the radar units. First, the radar calibration could drift. This value was checked about every three days and usually proved to be fairly stable. Second, the potentiometers used to attenuate the radar signal received by the computer could deviate from the desired set point. The attenuation was normally checked each time the radars were calibrated. It also proved to be quite stable. Third, the alignment of the radar antennas could be different than the required 45° angle. Since the radar only processed the velocity component directed at the face of the antenna and since a change in head alignment affected the elliptical size of the radar field of view, an error of 5° in alignment would result in an 8% error in apparent speed and a 14% error in apparent length. The majority of alignment errors occurred

during initial installation; however, since the antennas were mounted on sign bridges, structural vibration could affect alignment. The radar antennas were attached to safety rails bolted to the catwalks of the sign bridges. In order to combat vibration, guy wires were connected to the safety rails to aid in supporting the antennas. The final and most significant source of error in traffic measurement was attributed to the drift in the radar range setting. This setting affected the size of the field of view and consequently, the magnitude of the integrated signal. If the range was set too high, the radar would detect vehicles in adjacent lanes and would misfile the vehicles according to length. If the range was too low, the radar would occasionally not detect the smaller vehicles and would again misfile the traffic. Checks on the radar range settings were made each time the unit was calibrated or more frequently as needed. The process used to set the range was described in Chapter 5.

In order to maintain a high confidence level on the accuracy of the radars, manual counts of the traffic were frequently performed. Some typical values comparing radar data and manual counts are presented in Table 41. The error in measuring the total traffic volume was approximately -3.3% with a standard deviation of 4.3%. Since manual counts were usually performed just prior to calibration, the quoted error may be near the maximum error. The accuracy on the individual categories was not quite as high since no precise distinction could be made on some vehicles in each class. Furthermore, lane position was observed to affect the results on some of these border-line vehicles.

As a second check on the radar data, traffic information recorded by loop detectors in IH610 between N. Main St. and Airline Dr. were compared to radar data. The loop counter data consisted of hourly counts for both east and west bound directions. The data were supplied for the months of November and December in 1984, by the Texas State Department of Highways and Public Transportation. Since the loop counter data represented total vehicles per hour, the data were only compared to the total hourly radar counts. It would be impractical to compare loop counter results to manual counts since data would have to be counted for all lanes in one direction, simultaneously, for an entire hour.

Table 42 compares the radar and loop counter data. As can be seen for all monitoring periods, the radar data indicates a fewer number of total vehicles per hour than the loop counters. The loop counters simply tally the number of axles that pass over the detector loops. This total count is then essentially halved to represent an approximate passenger car equivalency. Since the radars gave a complete analysis of all vehicle fractions, one would expect the loop counters to indicate larger (and less accurate) vehicle counts.

The average weighted difference between the radars and loop counter data was -12%. This value was determined by taking the difference between the radar and loop counter totals and dividing this difference by the median of the two totals. The resulting value was then weighted

Table 41

Comparison between Radar Totals and Manual Counts

All Values in are Total Vehicles during the 5 Minute Interval

Date	Time	Radar	Category	Manual Count	Radar Count
11/21/84	1840–1845	1	Cat 1	12	10
			Cat 2	62	50
			Cat 3	0	5
			Cat 4	0	4
			Cat 5	0	1
Totals —				74	70

Date	Time	Radar	Category	Manual Count	Radar Count
11/21/84	1845–1850	3	Cat 1	13	20
			Cat 2	73	56
			Cat 3	1	5
			Cat 4	2	4
			Cat 5	0	2
Totals —				89	87

Table 41 (Continued)

Date	Time	Radar	Category	Manual Count	Radar Count
11/21/84	1850–1855	5	Cat 1	12	22
			Cat 2	63	48
			Cat 3	1	4
			Cat 4	2	2
			Cat 5	0	0
Totals —				78	76

Date	Time	Radar	Category	Manual Count	Radar Count
11/21/84	1855–1900	7	Cat 1	11	5
			Cat 2	37	42
			Cat 3	0	1
			Cat 4	1	1
			Cat 5	0	0
Totals —				49	49

Table 41 (Continued)

Date	Time	Radar	Category	Manual Count	Radar Count
11/21/84	1900–1905	9	Cat 1	3	6
			Cat 2	13	10
			Cat 3	1	0
			Cat 4	0	0
			Cat 5	0	0
Totals —				17	16

Date	Time	Radar	Category	Manual Count	Radar Count
11/21/84	2230–2235	2	Cat 1	5	9
			Cat 2	48	43
			Cat 3	0	1
			Cat 4	1	1
			Cat 5	0	0
Totals —				54	54

Table 41 (Continued)

Date	Time	Radar	Category	Manual Count	Radar Count
11/21/84	2225–2230	6	Cat 1	10	20
			Cat 2	44	29
			Cat 3	2	2
			Cat 4	0	0
			Cat 5	0	1
Totals —				56	52

Date	Time	Radar	Category	Manual Count	Radar Count
11/21/84	2225–2230	8	Cat 1	8	10
			Cat 2	20	16
			Cat 3	0	1
			Cat 4	0	0
			Cat 5	0	0
Totals —				28	27

Table 41 (Continued)

Date	Time	Radar	Category	Manual Count	Radar Count
11/21/84	2225–2230	10	Cat 1	0	4
			Cat 2	8	3
			Cat 3	0	0
			Cat 4	0	0
			Cat 5	0	0
Totals —				8	7

Date	Time	Radar	Category	Manual Count	Radar Count
11/29/84	2146–2151	1	Cat 1	5	15
			Cat 2	30	12
			Cat 3	0	2
			Cat 4	0	1
			Cat 5	0	0
Totals —				35	30

Table 41 (Continued)

Date	Time	Radar	Category	Manual Count	Radar Count
11/29/84	2146–2151	7	Cat 1	3	2
			Cat 2	13	14
			Cat 3	0	1
			Cat 4	1	0
			Cat 5	0	1
Totals —				17	18

Date	Time	Radar	Category	Manual Count	Radar Count
11/29/84	2151–2156	5	Cat 1	11	2
			Cat 2	28	38
			Cat 3	0	2
			Cat 4	2	1
			Cat 5	0	0
Totals —				41	43

Table 41 (Continued)

Date	Time	Radar	Category	Manual Count	Radar Count
11/29/84	2216–2221	6	Cat 1	3	10
			Cat 2	38	30
			Cat 3	2	1
			Cat 4	1	1
			Cat 5	0	0
Totals —				44	42

Date	Time	Radar	Category	Manual Count	Radar Count
11/29/84	2221–2226	2	Cat 1	8	9
			Cat 2	40	37
			Cat 3	0	1
			Cat 4	2	2
			Cat 5	0	0
Totals —				50	49

Table 41 (Continued)

Date	Time	Radar	Category	Manual Count	Radar Count
11/29/84	2221–2226	8	Cat 1	7	13
			Cat 2	16	8
			Cat 3	0	0
			Cat 4	0	0
			Cat 5	0	0
Totals —				23	21

Date	Time	Radar	Category	Manual Count	Radar Count
11/29/84	2221–2226	10	Cat 1	1	2
			Cat 2	6	1
			Cat 3	0	0
			Cat 4	0	0
			Cat 5	0	0
Totals —				7	3

Table 42

Comparison of Radar and Loop Counter Data

All Traffic Volumes are in Vehicles per Hour

Date	Hourly Period Ending	Radar Totals	Loop Totals	Percent Diff
11/15/84	1200	8068	9130	-12.4
	1300	7727	8780	-12.8
	1400	8028	9280	-14.5
11/21/84	1600	10093	12960	-24.9
	1700	11117	12550	-12.1
	1800	10469	11710	-11.2
	1900	8218	9070	-9.9
	2000	6735	7540	-11.3
	2100	5672	6280	-10.2
	2200	4809	5590	-15.0
11/29/84	1600	8223	11160	-30.3
	1700	8739	11620	-28.3
	1800	9700	11800	-19.5
	1900	7767	8910	-13.7
	2000	5600	6440	-14.0
	2100	4047	4810	-17.2
	2200	3857	4590	-17.4
12/04/84	0700	6704	8120	-19.1
	0800	9532	11240	-16.4
	0900	8418	10630	-23.2
	1000	7137	9530	-28.7
	1100	6769	9000	-28.3
	1200	6902	9230	-28.9
	1300	7281	9000	-21.1
	1400	7450	9290	-22.0
	1500	7952	9700	-19.8
	1600	9299	10940	-16.2
	1700	9989	11760	-16.3
	1800	10208	11480	-11.7
12/05/84	0800	10334	11750	-12.8
	0900	8676	10290	-17.0
	1000	7617	9220	-19.0

Table 42 (Continued)

Date	Hourly Period Ending	Radar Totals	Loop Totals	Percent Diff
12/05/84	1100	7180	9050	-23.0
	1200	7270	9420	-25.8
	1300	7622	9390	-20.8
	1400	7317	9330	-24.2
	1500	8174	9750	-17.6
	1600	9207	11090	-18.6
	1700	9784	11870	-19.3
	1800	10127	11710	-14.5
	1900	7088	8200	-14.5
	2000	5119	6010	-16.0
	2100	3212	4600	-35.5
12/06/84	1400	8282	9380	-12.4
	1500	8803	9960	-12.3
	1600	9843	11330	-14.0
	1700	10864	12310	-12.5
	1800	10776	12140	-11.9
	1900	7708	8660	-11.6
12/07/84	1100	7915	9120	-14.1
	1200	8555	9900	-14.6
	1300	8604	9780	-12.8
	1400	9075	10200	-11.7
	1500	9350	10680	-13.3
	1600	10418	11890	-13.2
	1700	10884	12410	-13.1
	1800	10783	12060	-11.2
	1900	8180	9220	-12.0
	2000	6828	7740	-12.5
	2100	5122	5800	-12.4
	2200	4403	5100	-14.7
	2300	3930	4620	-16.1
12/08/84	1200	6179	8590	-32.7
	1300	6809	8980	-27.5
	1400	6825	8640	-23.5
	1500	6966	8740	-22.6
	1600	6421	8580	-28.8
	1700	6840	8670	-23.6
	1800	6918	8660	-22.4

Table 42 (Continued)

Date	Hourly Period Ending	Radar Totals	Loop Totals	Percent Diff
12/10/84	0800	11147	12640	-12.6
	0900	9547	10660	-11.0
	1000	8000	8880	-10.4
	1100	7871	8740	-10.5
	1200	7755	9070	-15.6
	1300	7593	8870	-15.5
	1400	8311	9160	-9.7
	1500	8758	9690	-10.1
	1600	9925	11050	-10.7
	1700	10652	11860	-10.7
	1800	10295	12040	-15.6
	1900	7357	8490	-14.3
	2000	5583	6120	-9.2
	2100	4015	4470	-10.7
12/11/84	0700	7825	9520	-19.5
	0800	11201	12700	-12.5
	0900	9269	10890	-16.1
	1000	8475	9380	-10.1
	1100	8344	9280	-10.6
	1200	8281	9150	-10.0
	1300	7774	9090	-15.6
	1400	8357	9290	-10.6
	1500	8804	9850	-11.2
	1600	10169	11330	-10.8
	1700	10279	11850	-14.2
	1800	10725	12110	-12.1
12/12/84	0800	11720	13160	-11.6
	0900	9659	10580	-9.1
	1000	8182	9130	-11.0
	1100	7905	8700	-9.6
	1200	8100	9120	-11.8
	1300	8133	8970	-9.8
	1400	8334	9290	-10.8
	1500	8590	9520	-10.3
	1600	9172	10710	-15.5
	1700	10227	12030	-16.2
	1800	10765	11860	-9.7
12/13/84	0900	9730	10910	-11.4
	1000	8434	9410	-10.9

Table 42 (Continued)

Date	Hourly Period Ending	Radar Totals	Loop Totals	Percent Diff
12/13/84	1100	7762	8980	-14.6
	1200	8546	9460	-10.2
	1300	8158	9150	-11.5
	1400	8297	9740	-16.0
	1500	8927	9850	-9.8
	1600	10119	11410	-12.0
	1700	10540	12470	-16.8
	1800	10630	11890	-11.2
	1900	8220	9040	-9.5
12/18/84	1300	8595	9640	-11.5
	1400	8723	9810	-11.7
	1500	9231	10440	-12.3
12/19/84	1000	7663	9550	-21.9
	1100	7587	9750	-25.0
	1200	8258	10090	-20.0
	1300	8123	9610	-16.8
	1400	8485	10040	-16.8
	1500	8940	10530	-16.3
	1600	9460	11180	-16.7

by the median to determine the average weighted difference. Since the radar error on total vehicle counts was established as approximately −3.3% (the radars indicated an average of 3.3% fewer vehicles than the true total) the loop counters were then probably about 9% higher than the true count.

It should be stated that one would expect the radars to indicate fewer total vehicles than the true total. This can be attributed to vehicle tailgating. If the distance between adjacent vehicles was smaller than the length of the major axis of the elliptical radar beam, the radars would count the two vehicles as one long vehicle. This phenomenon would be a maximum at periods of peak traffic flow.

B. Establishment of an Air Quality Data Base

A sizable quantity of data were collected in the experimental phase of project 283. These data included measurements of traffic volumes, meteorological conditions, and pollutant concentrations. An overview of the data base is given below.

Qualitative Discussion of the Data Base

Data were collected on magnetic tape in a form that allows for realization of the dynamic response in the instrumentation. Instantaneous values for all instruments on the north side of the freeway (normally downwind) were recorded at a rate equal to the user supplied sample rates. Even though only five minute and hourly averages are available for the south side of the freeway, these data should suffice for most developmental work since these were primarily upwind readings.

Included in the pollutant data are concentrations of carbon monoxide, oxides of nitrogen, ozone, and hydrocarbons. Only carbon monoxide data is available for more than one receptor height. Usually only one upwind carbon monoxide receptor could be monitored due to lack of functioning instruments.

Wind speed and direction for all receptor heights on both towers were recorded on tape. Additional meteorological instruments were placed at locations listed in Tables 17–18. Care should be exercised when working with the wind directions recorded by the DART on the south tower. The software used by the DART was not capable of properly calculating the wind direction averages during prevailing northerly winds. This is because a simple arithmetic mean may not be used to calculate the average wind direction. However, when the winds were northerly, the UVW anemometers on the same tower should give good values of the wind direction observed on the south tower. Since all UVW anemometer data were collected by the Balcones computer and this computer recorded discrete data, the wind directions reported for all UVW anemometers could properly be calculated in the data reduction process.

An accurate traffic data base was consequently collected by the computer. The traffic data included breakdowns on volume, size, and speed on a lane-by-lane basis. A detailed breakdown

of the entire data base format is presented in in Appendix G. Fifteen minute and hourly averages acquired in Houston are presented in Appendix K of this report. (Appendix K is bound separately.)

Quantitative Discussion of the Data Base

Since it would be impractical to tabulate all of the experimental data gathered during the experimental phase of the project, a brief discussion of observed pollutant concentrations will be presented. Table 43 illustrates the maximum pollutant concentrations observed during the data acquisition. Table 44 presents the National Air Quality Primary Pollutant Standards applicable at the time.

The maximum concentrations given in Table 43 contain two sets of averages. The first average consists of the maximum five minute concentrations during each day when the wind was from the south. The second set of averages consists of the maximum hourly concentrations observed during each day. Again, only maximums for days in which the wind was from the south are presented. This is due to the fact that the majority of monitoring instrumentation was located on the north side of the roadway, or downwind of the roadway during southerly winds. Furthermore, when more than one instrument was used to monitor a specific pollutant, (*e.g.,* carbon monoxide), the maximum reported concentration arises from the monitor giving the highest level during the day.

Table 43 shows that all pollutant concentrations were relatively low at all times except around December 10, 1984. At about 6:00 p.m. Friday, December 7, 1984, all of the analyzers began to show higher pollutant concentrations than had previously been observed. These high concentrations continued through about noon on Wednesday, December 12. During the morning rush hour of Monday, December 10, the pollution concentrations peaked. Nitrogen oxide levels were so high that the analyzers were not able to report the concentration on the normal range selected for ambient monitoring.

During the preparation of the research site, the DASIBI carbon monoxide and ozone analyzers were left in their monitoring positions. Ozone concentrations were observed to surpass the standard on several occasions especially during the mid-summer months. However, since these concentrations occurred during site preparation, they were not permanently recorded. Carbon monoxide levels were never observed to exceed about 6 ppm during the same period. The highest concentrations of CO and most other pollutants measured in Houston during the entire time were those occurring during the few days surrounding December 10.

C. Emission Factor Estimation

Every dispersion model must have the source strength to estimate concentrations at various receptors. This strength may be internally calculated (as in TEXIN), or the user must supply the required strength. For roadway dispersion models, the source strength is usually expressed on a composite mass per length per vehicle basis. This characterization leads to an emission factor.

Table 43

Maximum Sustained Pollutant Concentrations

Concentrations in ppm

Maximum Sustained Five Minute Concentrations

Date	NO_x	NO_2	THC	NMHC	CO_2	CO	O_3
08/08/84	—	—	—	—	—	3.5	—
08/23/84	—	—	—	—	—	4.4	—
08/24/84	—	—	—	—	—	4.2	—
11/15/84	—	—	—	—	—	4.1	.026
11/29/84	—	—	2.8	3.0	416	3.6	.023
12/07/84	—	—	5.6	4.6	553	10.8	.033
12/08/84	.255	.030	2.7	—	411	3.7	.038
12/10/84	> .5†	.050†	6.4	6.3	1050	11.0	.026
12/11/84	.497	.033	6.5	5.6	597	9.4	.029
12/12/84	.314	.033	3.2	4.0	956	4.8	.030
12/13/84	.173	.030	2.4	4.7	491	3.1	.044
12/18/84	—	—	—	—	—	3.0	.025

Maximum Sustained Hourly Concentrations

Date	NO_x	NO_2	THC	NMHC	CO_2	CO	O_3
08/08/84	—	—	—	—	—	3.4	—
08/23/84	—	—	—	—	—	4.1	—
08/24/84	—	—	—	—	—	3.2	—
11/15/84	—	—	—	—	—	2.2	.020
11/29/84	—	—	2.6	2.7	407	3.2	.018
12/07/84	—	—	5.0	4.2	529	7.6	.022
12/08/84	.200	.025	2.2	—	399	3.0	.031
12/10/84	> .426†	.040†	5.7	5.4	776	8.0	.022
12/11/84	.444	.030	5.1	4.8	519	8.0	.024
12/12/84	.270	.027	3.0	3.5	488	4.0	.025
12/13/84	.142	.025	2.2	3.0	420	2.8	.022
12/18/84	—	—	—	—	—	2.5	.021

†These values represented measurements in which the respective monitors showed a concentration out of their rated monitoring ranges. The exact values are therefore suspect, but are presented here for comparison. The true readings are probably somewhat higher than the illustrated values.

Table 44

National Ambient Air Quality Standards[67]

Primary Standards†

Revised 2/1/85

Pollutant	Standard
Carbon Monoxide	(a) 9 ppm (10 mg/m^3) maximum 8 hr. concentration not to be exceeded more than once per year (b) 35 ppm (40 mg/m^3) maximum 1 hr. concentration not to be exceeded more than once per year
Oxides of Nitrogen (NO_2)	0.05 ppm (100 μg/m^3) annual arithmetic mean
Ozone (O_3)	0.12 ppm (235 μg/m^3) expected daily violations averaging less than once per year over a three year period
Suspended Particulate Matter	75 μg/m^3 annual geometric mean 260 μg/m^3 maximum 24 hr. concentration
Sulfur Dioxide (SO_2)	0.03 ppm (80 μg/m^3) annual average 0.14 ppm (365 μg/m^3) maximum 24 hr. concentration
Lead (Pb)	1.5 μg/m^3 average over a calendar year

†Levels of air quality necessary to protect the public health with adequate margins of safety.

In order to quantify the composite emission factors in the experimental environment, the mass balance procedure was performed.[68] This procedure calculates the mass flux of a species as it passes a downwind receptor. Only the component of the wind speed flowing normal to the roadway is used to calculate the flux. This insures that the contribution of the pollutant from the road passing the receptor may be calculated from a knowledge of upwind concentration. In general, the calculational procedure assumes that the amount of material flowing past a downwind receptor minus the amount of material flowing past an upwind receptor is due to the roadway, assuming no sinks or decomposition of material between the upwind and downwind receptors. A sample mass balance calculation is given in Appendix H.

Errors in the Calculational Procedure

Since the technique utilizes virtually all types of data obtained by the project, the errors inherent in performing the balance will be compounded by measurement errors. This error will be a maximum when all instruments present their maximum errors simultaneously. Even though the probability of this occurring is small, a few points should be taken into consideration in order to keep the errors to a minimum.

The errors in measuring the wind direction had a large influence on the mass balance. Since the sine of the angle with respect to the road is used to compute the component of the wind normal to the road, the error associated with the wind vanes will have the greatest effect at small angles where the sine function has the steepest slope. Therefore, only cases where the wind angle with respect to the roadway was greater than 20° were considered. At a 20° angle, a 5° error results in a 20% change in emissions calculated by the mass balance technique. However, at 45°, the same error results in only an 8.3% change in calculated emissions. These sensitivity estimates assume that four downwind direction measurements were in error by the maximum amount at the same time.

The wind speed was accurate to within about 0.5 mile per hour. Since low wind speed cases were not considered in estimating the emission factors, the maximum effect on the mass balance results would be about 17%.

The error in concentration measurement would have the greatest effect when the upwind values were about the same as the downwind rates. However, in most cases when the technique could be adequately applied, the concentration differences between upwind and downwind receptors were significant (2.0 ppm or greater for CO). Since different cylinders of span gas were used to calibrate the upwind instruments and the downwind instruments, inconsistent analysis of calibration standards could also play an important role. In order to limit this effect, the standards were all checked in the same instrument before they were used for calibration purposes.

The source strength was converted to a per vehicle basis from a knowledge of the traffic volume

on the roadway during the interval. Since the radars were accurate to normally 5% or better, the errors in traffic measurement could introduce a 5% error. This is due to the fact that the total emissions were simply divided by the traffic volume to obtain the emission factor.

CO Emission Factor Results

The mass balance technique was applied to the data base for all cases that met the criteria in the above section. The process is illustrated by a sample calculation in Appendix H. Table 45 lists the emission factors calculated by the mass balance technique (Exp) and MOBILE3. Several different MOBILE3 cases are presented and will now be discussed.

When an emission model such as MOBILE3 is normally invoked, the VMT (vehicle miles traveled) mix for the area that is supplied to the model usually is obtained from registration data. This process was used to establish the registration emission factors given in the table (SDHPT scenario). The data were obtained from the latest available registration data for Harris County, Texas, from the Texas State Department of Highways and Public Transportation (SDHPT). The various vehicle classes significant in the MOBILE3 vehicle scenario are:

LDGV: Light duty gasoline vehicles

LDGT1: Light duty gasoline trucks with a gross vehicle weight rating (GVWR) less than 6001 lbs

LDGT2: Light duty gasoline trucks with a GVWR less than 8501 lbs

HDGV: Heavy duty gasoline powered vehicles

LDDV: Light duty Diesel vehicles

LDDT: Light duty Diesel trucks

HDDV: Heavy duty Diesel vehicles

MC: Motorcycles

The registration data were further used to establish a distribution of vehicle age in the Houston area.

The TTI scenario data in Table 45 result from an estimation of the VMT mix obtained from the radar information. Since the vehicle classification scheme used by MOBILE3 did not agree with the classification scheme used by the radars, estimates for the percentage of the various MOBILE3 classes in each radar category were used to obtain the vehicle scenario. However, the same vehicle registration distribution data were used in both the registration and experimental cases. Parameters used in establishing the VMT mix are presented in Tables 46–47 and the vehicle registration distribution is given in Table 48. In the execution of MOBILE3, the VMT mix for the year 1984 was used but the 1983 registration distribution data had to be used since it was the latest available data at the time.

Data supplied by the SDHPT gave the values of the percent of VMT accumulated in the

Table 45

MOBILE3 and Mass Balance CO Emission Factors

All Emission Factors are in gm/vehicle · mile

Date	5 Min Period Ending	Exp	MOBILE3 SDHPT Scenario[†]	MOBILE3 TTI Scenario[‡]
11/15/84	1410	34.7	12.9	21.3
	1420	38.6	13.4	20.1
	1425	28.2	14.0	20.6
	1430	10.2	13.3	21.3
	1435	7.2	14.0	22.1
	1450	42.9	12.5	23.4
11/29/84	1525	23.3	10.6	19.6
	1530	26.9	9.8	17.7
	1535	35.0	9.9	17.9
	1540	34.3	10.5	17.6
12/07/84	1400	9.1	10.7	15.8
	1420	8.5	10.9	18.3
	1510	15.4	10.7	19.1
	1745	28.9	11.9	15.2
	1805	30.3	11.8	15.8
	1820	29.8	12.0	14.7
	1825	36.4	11.7	14.0
	1835	15.7	11.8	14.3
	1840	10.3	11.7	14.2
	1845	36.6	12.1	14.8
	1950	18.6	12.2	14.2
	2030	24.5	12.2	13.1
	2040	16.6	14.9	16.1
	2045	16.2	14.7	14.3
12/10/84	1110	8.4	10.1	15.5
	1220	11.8	10.6	18.0
	1225	18.8	10.4	16.0
	1315	15.1	9.9	16.8
	1400	26.7	10.1	17.2
	1405	22.3	10.6	16.8
	1410	31.0	11.1	18.9
	1415	25.9	10.3	17.4
	1420	30.3	10.5	18.5

Table 45 (Continued)

Date	5 Min Period Ending	Exp	MOBILE3 SDHPT Scenario[†]	MOBILE3 TTI Scenario[‡]
12/10/84	1425	20.9	10.5	16.6
	1430	14.2	10.1	17.9
	1445	34.1	10.7	17.6
	1450	35.2	10.4	17.3
	1455	30.9	10.3	17.8
	1500	40.1	10.8	19.1
	1505	14.5	10.5	16.1
	1650	15.4	10.4	15.5
	1715	20.8	10.3	13.6
	1830	41.2	10.1	13.1
	1835	40.2	10.1	14.1
	1845	21.8	10.1	14.3
	1850	46.5	10.2	14.1
	1855	29.0	10.4	11.5
	1900	29.3	10.1	13.6
12/11/84	0655	28.5	10.9	16.0
	1005	33.1	10.0	16.4
	1810	24.4	10.3	13.5
12/12/84	0635	13.5	11.1	16.4
	0640	26.5	11.0	16.4
	0645	15.4	11.1	17.0
	0650	26.8	11.0	15.6
	0655	18.9	10.9	14.7
	0700	18.9	10.5	14.8
	0705	20.5	10.6	15.8
	0710	23.8	10.8	15.7
	0720	25.4	10.7	16.4
	1405	28.7	10.5	18.1

[†]These data result from the use of registration data to obtain the VMT mix.
[‡]These data result from the use of radar information to *estimate* the VMT mix.

Table 46

Registration VMT Mix[†]

Harris County, Texas

Year	LDGV[a]	LDGT1[b]	LDGT2[c]	HDGV[d]	LDDV[e]	LDDT[f]	HDDV[g]	MC[h]
1980	.747	.126	.081	.022	.004	.001	.012	.007
1981	.743	.127	.082	.020	.007	.001	.012	.008
1982	.736	.129	.083	.020	.010	.002	.012	.008
1983	.731	.132	.084	.018	.012	.003	.012	.008
1984	.724	.136	.084	.017	.013	.005	.013	.008
1985	.718	.141	.083	.015	.016	.006	.013	.008
1986	.710	.146	.082	.014	.019	.007	.014	.008
1987	.705	.152	.079	.013	.020	.009	.014	.008
1988	.696	.159	.076	.013	.023	.010	.015	.008
1989	.691	.164	.072	.012	.025	.012	.016	.008
1990	.682	.174	.069	.011	.026	.014	.016	.008
1991	.683	.174	.066	.010	.028	.015	.016	.008
1992	.676	.179	.063	.010	.030	.017	.017	.008
1993	.674	.183	.059	.009	.031	.019	.017	.008
1994	.667	.187	.057	.009	.033	.021	.018	.008
1995	.666	.190	.054	.008	.033	.023	.018	.008
1996	.657	.196	.052	.008	.037	.024	.018	.008
1997	.652	.198	.050	.008	.038	.027	.019	.008
1998	.647	.200	.048	.007	.041	.030	.019	.008
1999	.644	.204	.046	.007	.041	.031	.019	.008
2000	.641	.205	.045	.007	.042	.033	.019	.008

[†]All data later than 1983 are projected values.

[a]Light Duty Gas Vehicles
[b]Light Duty Gas Trucks (GVWR < 6001 lbs)
[c]Light Duty Gas Trucks (GVWR < 8501 lbs)
[d]Heavy Duty Gas Vehicles
[e]Light Duty Diesel Vehicles
[f]Light Duty Diesel Trucks
[g]Heavy Duty Diesel Vehicles
[h]Motorcycles

Source: Texas State Department of Highways and Public Transportation

Table 47

TTI Estimates of the VMT Mix

MOBILE3 Class	Estimated Percentage
LDGV	Cat 1 + 77% Cat 2 + 50% Cat 5
LDGT1	15% Cat 2
LDGT2	15% Cat 3 + 5% Cat 2
HDGV	85% Cat 3
LDDV	1% Cat 2
LDDT	2% Cat 2
HDDV	Cat 4 + 50% Cat 5
MC	0%

Table 48

Vehicle Registration Distribution

Harris County, Texas

Model	LDGV	LDGT1	LDGT2	HDGV	LDDV	LDDT	HDDV	MC
1983	.065	.068	.076	.033	.065	.068	.036	.133
1982	.083	.087	.098	.057	.083	.087	.044	.145
1981	.098	.112	.126	.104	.098	.112	.085	.138
1980	.097	.095	.107	.105	.097	.095	.126	.116
1979	.085	.067	.075	.101	.085	.067	.093	.123
1978	.099	.093	.104	.125	.099	.093	.118	.114
1977	.097	.086	.097	.100	.097	.086	.098	.069
1976	.084	.077	.083	.075	.084	.077	.103	.044
1975	.069	.059	.061	.047	.069	.059	.047	.024
1974	.044	.036	.036	.046	.044	.036	.056	.009
1973	.043	.041	.036	.047	.043	.041	.049	.085
1972	.037	.036	.028	.041	.037	.036	.045	.000
1971	.026	.028	.019	.028	.026	.028	.029	.000
1970	.020	.024	.015	.018	.020	.024	.017	.000
1969	.015	.020	.011	.010	.015	.020	.009	.000
1968	.011	.017	.008	.008	.011	.017	.007	.000
1967	.008	.014	.006	.007	.008	.014	.006	.000
1966	.006	.010	.004	.006	.006	.010	.005	.000
1965	.005	.008	.003	.005	.005	.008	.004	.000
1964	.008	.022	.007	.037	.008	.022	.023	.000

Source: Texas State Department of Highways and Public Transportation

cold start mode by non-catalyst equipped vehicles (PCCN), in the hot start mode by catalyst equipped vehicles (PCHC), and in the cold start mode by catalyst equipped vehicles (PCCC) for the entire county. Since the experimental site represented a freeway driving situation only, more conservative estimates of PCCN, PCHC, and PCCC were chosen. These values were estimated from the supplied hourly breakdowns of PCCN, PCHC, and PCCC which are given in Table 49. Final values of PCCN, PCHC, and PCCC were estimated by dividing the county-wide hot/cold start factors by two.

Since data on relative humidity and temperature were known for each mass balance case, air conditioning usage factors could be estimated by MOBILE3 and their influence applied to the emission factors. To effect this procedure, the wet bulb temperature had to be determined from the relative humidity and temperature. An empirical relation that gave relative humidity as a function of wet and dry bulb temperatures, as well as atmospheric pressure was used. Since this function was non-linear in wet bulb temperature, Newton's method was used to find the zero of the function.

Anti-tampering programs (ATP) were used in MOBILE3. These ATP programs were approved for use by the Texas Air Control Board. When ATP programs are used in MOBILE3, three different cases must be modeled. First, the emissions model was executed without an ATP. Secondly, an ATP was employed to cover the 1968–1979 models. This ATP included annual inspections of the air pump, evaporative canister, and the pollution control valve (PCV). Finally, an ATP was used to cover the 1980–2020 year models. The last ATP covered annual inspection of the air pump, catalyst, fuel inlet, evaporative canister, and PCV. For all cases, the emission factors calculated by MOBILE3 were the same whether an ATP program was invoked or not for the 1984 scenario. Therefore, the numbers given in Table 45 represent emission factors for all MOBILE3 cases. If the calendar year being modeled by MOBILE3 is increased to 1990 or later, the anti-tampering programs begin to have an effect. However, since this year was not of interest in comparing MOBILE3 estimates to mass balance calculations, it is not presented here.

Table 45 indicates that in virtually all cases, the emission factors estimated by MOBILE3 were lower than those estimated by the mass balance technique. Furthermore, the estimated vehicle scenario obtained from the radars normally gave much better agreement with the mass balance result. These comparisons allow one to conclude that a major source of error in dispersion modeling may arise from incorrect assumptions in emission models used to calculate emission factors.

D. SF_6 Tracer Gas Experiments

Tracer gas experiments were performed on the last two monitoring days in Houston. The experiments were attempted in order to compare mass balance results to a known tracer emission rate. Air samples were taken by syringe samplers over 15 minute intervals while the tracer was

Table 49

Hourly Summary of Hot/Cold Start Factors

Harris County, Texas

Hour	PCCN[a]	PCHC[b]	PCCC[c]
00	21	7	28
01	22	7	28
02	23	7	28
03	24	7	27
04	26	7	27
05	25	7	26
06	25	8	26
07	24	10	25
08	18	12	20
09	13	16	18
10	10	18	18
11	8	19	17
12	8	19	17
13	8	18	18
14	10	18	18
15	13	17	19
16	14	18	20
17	13	15	19
18	10	17	19
19	8	17	20
20	6	16	21
21	7	13	24
22	11	10	25
23	17	9	26

[a]Percent cold start non-catalyst vehicles
[b]Percent hot start catalyst vehicles
[c]Percent cold start catalyst vehicles

Source: Texas State Department of Highways and Public Transportation

being released.

The mass balance technique for determining the SF_6 observed emission rate was applied as in the carbon monoxide data for all cases where the concentration profile was adequately defined. A sample calculation using SF_6 is included in Appendix H. Appendix I gives the tracer gas concentrations observed during the experiment and the corresponding meteorology. Table 50 lists the emission rates calculated by the mass balance technique and the actual tracer release rate. Since the amount of data that were suitable for mass balance analysis was limited, no strong conclusions can be drawn from the information. However, since the observed emission rate is consistently smaller than the measured rate, a few possible defects in the experimental procedure will be discussed.

Due to the cost of the procedure, a well designed and organized tracer gas experiment was not conducted. In order to cut expenses, sacrifices in the tracer gas experiment were made.

The first day the tracer gas experiment was conducted, the concentration of SF_6 at the highest downwind receptors was much higher than desired. For safety purposes, the samplers were only lifted to a height of 47 feet on that day. The observed concentrations at the highest receptors indicated that the tracer plume was probably moving over the top receptors. Therefore, on the second day, the highest samplers were lifted to a height of 59 feet.

In order to receive a better response from the chromatograph used to analyze the air samples, it was decided to increase the emission rate of SF_6 on the second day. This larger flow rate gave a much stronger response and possibly may have been too strong. A tracer emission rate between the two values would be a good choice.

The last two sampling periods (30 min) on the second day consider the dispersion of the tracer once the source has been extinguished. These two periods illustrated that the concentrations were still quite high even up to 30 minutes after discontinuing tracer emission. Hence, successive experiments should be sufficiently spaced in order to disperse as much of the tracer as possible from the area before beginning additional runs. Furthermore, the concentration profiles were essentially reversed during these two intervals (*i.e.*, the concentration of SF_6 was higher near the top of the tower).

The difference between calculated emission rates using the mass balance technique and actual tracer emission rates can be attributed to the method used to emit the tracer. The mass balance technique assumes a uniform continuous source. Periods when the tracer was not being emitted on the freeway, *i.e.*, waiting at an intersection to turn around, would drastically decrease the observed apparent emission rate. Since it was feasible for the emitting vehicles to spend one-half of the time at the intersections, that waiting period may have had a dramatic effect on the data.

A suggested method to use in tracer emission involves more vehicles and the use automated control valves to regulate the tracer flow. The solenoid valves could then be used to stop the emission at the turn around areas in order to minimize end effects. The valve would then be

Table 50

Tracer Gas Emission Rate Comparisons

All Emission Rates are in gm SF_6/min

Date	Time Interval	Mass Balance Rate	Actual Rate
12/18/84	1309–1324	1.5	6.0
12/18/84	1324–1339	1.9	6.0
12/19/84	1137–1152	7.6	17.6
12/19/84	1152–1207	6.8	17.6
12/19/84	1207–1222	12.9	17.6
12/19/84	1222–1237	5.1	17.6
12/19/84	1408–1423	15.6	17.8
12/19/84	1423–1438	17.1	17.8

opened once the vehicle was ready to again begin traveling across the research area. The total time that the solenoid valve was open would be recorded as well as the total time of sampling period. This would enable a correction due to total emission time being less than sample time. A longer distance should also be traveled by the vehicles between turn-arounds in order to maximize the time spent emitting the tracer as a *line* source. A distance of about 900 meters was used at the Houston site.

It is further recommended that the highest receptor be located as high as possible on the downwind side. This would keep the plume from rising above the top downwind receptor. Verification of the mass balance technique was done by Bullin, *et al.*[11] for the GM and SRI data bases. It was shown that many of the mass balance cases were in good agreement with the actual emission rates. The cases sited indicate that the results from the tracer gas experiments under this project were probably in error due to experimental procedure problems.

E. Methods to Improve CO Emission Factor Estimation

The best way to improve the estimation of carbon monoxide emission factors from the mass balance technique would be to improve the instrumentation used to gather the samples. Since the accuracy of the mass balance technique depends on the accuracy of the carbon monoxide analyzers used, more precise instruments should be used instead of Ecoloyzers if available. The DASIBI monitors would be an excellent example.

In order to better characterize the concentration profile, more receptors would be desirable. From the available data, it was realized that considering the project layout, the receptor at the top of the downwind tower (102 ft) was quite important. Hence, more receptors should be distributed between ground level and the top of the tower. It is also desirable to have more upwind receptors in order to better characterize those data also.

Chapter 8

Conclusions and Recommendations

Project 283 considered both model development and experimental work in air quality research near roadways. The TXLINE model was modified to model the roadway as a finite source rather than an infinite line source. The new model, TXLINE-2, has the added capability of predicting pollutant concentrations upwind from roadways. The TXLINE-2 model was statistically compared to several existing air pollution data bases. Low wind speed correction factors were added to improve the performance under these conditions. The model can be used to predict pollutant concentrations downwind of a singular finite line source or several parallel finite line sources of any elevation.

The original TEXIN model, which was previously developed to predict carbon monoxide concentrations near intersections, had several restrictions which inhibit its use in many general cases. The model was expanded to include modeling capabilities of four-way stop intersections. The addition of the CMA Operations and Design traffic algorithm allowed for more detailed intersection treatment by the application of several adjustment factors. T-intersections are now specifically treated by a special model flag. The EPA emissions model MOBILE3, enables the user to more accurately estimate source strength. Furthermore, MOBILE3 includes many new calculation methodologies and algorithms including vehicle anti-tampering and inspection/maintenance programs to enhance its flexibility. A short-cut emissions calculational procedure has been added to TEXIN for those who do not wish to use the rather large MOBILE3 routine.

A sizable quantity of experimental data at a Houston at-grade research site was collected under the project. These data included traffic volumes, meteorological parameters, and pollutant concentrations at several receptor locations. Observed pollutant concentrations at the research site were normally relatively low except for a period of a few days. The mass balance technique was applied to the carbon monoxide data to calculate experimental emission factors. These experimental emission factors were then compared to predicted emission rates generated by MOBILE3 for two different scenarios. The experimental emission factors were virtually always higher than those predicted by the MOBILE3 model. A sulfur hexafluoride tracer gas experiment was performed in Houston and the mass balance technique was also applied to those results.

The data collected in Houston was arranged in a manner so that it can be used for subsequent model evaluation and/or development. These data reside on nine-track magnetic tape in a form described in Appendix G.

Recommendations

Future research work in this area should consider the following:

(1) Determination of the effects of induced vehicle turbulence on the dispersion process.

(2) Performing additional tracer gas experiments in order to better verify the mass balance technique comparison to MOBILE3.

(3) The use of more accurate carbon monoxide analyzers.

(4) The use of more receptors both upwind and downwind of the roadway.

(5) Improvement of the overall accuracy of the TXLINE-2 model. TXLINE-2, like all line source models investigated in this study, tends to underpredict the experimental data.

(6) Improvements in the TEXIN2 and TXLINE-2 models in order to reduce scatter in the statistical analyses.

(7) The detailed investigation of the dispersion process at low wind speeds.

(8) The development of street canyon options for TEXIN2 and the possible use of CALINE4 to model the dispersion in TEXIN2.

References

References

1. J. L. Beaton, E. C. Shirley, J. B. Skog, "A Method of Analyzing and Reporting Highway Impact on Air Quality," U. S. DOT, Federal Highway Administration, FHWA-RD-72-39, April 1972.

2. J. A. Bullin, J. B. Rodden, A. D. Messina, "TXLINE: A Computer Model for Estimating Pollutant Concentrations Downwind of a Roadway," Texas Transportation Institute Research Report 283-1, Texas A&M University, College Station, TX, 1983.

3. P. E. Benson, "CALINE4—A Dispersion Model for Predicting Air Pollutant Concentrations Near Roadways," California Department of Transportation, Sacramento, California, 1984.

4. W. B. Petersen, "User's Guide for HIWAY-2, A Highway Air Pollution Model," Environmental Sciences Research Laboratory, U. S. EPA, Research Triangle Park, N. C., EPA-600/8-80-018, 1980.

5. A. D. Messina, J. A. Bullin, J. P. Nelli, R. D. Moe, "Estimates of Air Pollution Near Simple Signalized Intersections," U. S. DOT, Federal Highway Administration, FHWA/RD-83/009, 1983.

6. "Carbon Monoxide Hot Spot Guidelines—Volume V: User's Manual for Intersection Midblock Model," U. S. EPA, Office of Air Quality Planning and Standards, Research Triangle Park, N. C., EPA-450/3-78-037, 1978.

7. R. Griffin, "Air Quality Impact of Signaling Decisions," U. S. DOT, Federal Highway Administration, FHWA/CO/RD-80/12, 1980.

8. R. Griffin, "Air Quality Impact of Signaling Decisions—Program MICRO User's Guide," U. S. DOT, Federal Highway Administration, FHWA/CO/RD-80/13, 1980.

9. "Compilation of Air Pollutant Emission Factors," Office of Air Quality Planning and Standards, U. S. EPA, Research Triangle Park, N. C., Publ. AP-42, 1975.

10. P. Kunselman, H. T. McAdams, C. J. Domke, M. Williams, "Automobile Exhaust Emission Modal Analysis Model," U. S. EPA, Office of Air and Water Programs, Ann Arbor, MI, EPA-460/3-74-005, 1974.

11. J. A. Bullin, J. C. Polasek, N. J. Green, "Analytical and Experimental Assessment of Highway Impact on Air Quality," Texas Transportation Institute Research Report 218-4, Texas A&M University, College Station, TX, 1978.

12. "User's Guide to MOBILE3: Mobile Source Emissions Model," Office of Air and Radiation, Office of Mobile Sources, Emission Control Technology Division, U. S. EPA, Ann Arbor, MI, 1984.

13. Fong-Ping Lee, C. E. Lee, R. B. Machemehl, C. R. Copeland, "Vehicle Emissions at Intersections," Center for Transportation Research Report 250-1, University of Texas at Austin, Austin, TX, 1983.

14. C. E. Lee, G. E. Grayson, C. R. Copeland, J. W. Miller, T. W. Rioux, V. S. Savur, "The Texas Model for Intersection Traffic—Programmer's Guide," Center for Highway Research, University of Texas at Austin, Report No. 184-3, Austin, TX, 1977.

15. Hsin Hsing Wu, "Modeling Heavy-Duty Gasoline Vehicle Emissions and Fuel Consumption," M. S. Thesis, College of Engineering, Department of Civil Engineering, University of Texas at Austin, Austin, TX.

16. F. Pasquill, *Atmospheric Diffusion*, 2nd ed., Ellis Horwood Ltd., Sussex, England (1974).

17. O. G. Sutton, *Micrometeorology*, Mc-Graw Hill Book Company, New York (1953).

18. S. R. Hanna, G. A. Briggs, R. P. Hosker, "Handbook on Atmospheric Diffusion," Office of Health and Environmental Research, U. S. DOE, DOE/TIC-11223, 1982.

19. R. E. Treybal, *Mass Transfer Operations*, 2nd ed., Mc-Graw Hill Chem. Eng. Series, Mc-Graw Hill Book Company, New York (1968).

20. F. B. Smith, "The Diffusion of Smoke from a Continuous Elevated Point-Source into a Turbulent Atmosphere," *J. Fluid Mech.*, **2**:49 (1957).

21. N. J. Green, "Roadway Pollutant Dispersion: Development of a Data Base and a Model Evaluation of Five Models," M. S. Thesis, Texas A&M University, College Station, TX, 1980.

22. H. E. Cramer, "Engineering Estimates of Atmospheric Dispersal Capacity," *American Industrial Hygiene Association Journal*, **20**:183 (1959).

23. K. L. Calder, "Eddy Diffusion and Evaporation Over Aerodynamically Smooth and Rough Surfaces: A Treatment Based on Laboratory Laws of Turbulent Flow with Special Reference to Conditions in the Lower Atmosphere," *Quart. J. Mech. and Applied Math.*, **2**:153 (1949).

24. M. L. Barad, "Project Prairie Grass; A Field Program in Diffusion," Vol. II, Geophys. Research Papers No. 59, (1958).

25. F. A. Gifford, "Turbulent Diffusion Typing Schemes—A Review," *Nucl. Saf.*, **17**:68 (1976).

26. F. Pasquill, "The Estimation of the Dispersion of Windborne Material," *Met. Mag.*, **90**:33 (1961).

27. J. R. Zimmerman, R. S. Thompson, "User's Guide for HIWAY: A Highway Air Pollution Model," U. S. EPA, Research Triangle Park, N. C., EPA-650/4-74-008, 1975.

28. D. B. Turner, "Workbook of Atmospheric Dispersion Estimates," Air Resources Field Research Office, Environ. Science Services Administration, Public Health Service Publication No. 999-AP-26, Cincinnati, OH, 1970.

29. S. T. Rao, M. Keenan, F. Sistla, P. Sampson, "Dispersion of Pollutants Near Highways—Data Analysis and Model Evaluation," U. S. EPA, Research Triangle Park, N. C., EPA-600/4-79-011, 1979.

30. R. E. Eskridge, J. C. Hunt, "Highway Modeling: Part I— Prediction of Velocity and Turbulence Fields in the Wake of Vehicles," *J. Appl. Meteorology*, **36**:387 (1979).

31. R. E. Eskridge, F. S. Binkowski, J. C. R. Hunt, T. L. Clark, K. E. Demerjian, "Highway Modeling: Part II—Advection and Diffusion of SF_6 Tracer Gas,", *J. Appl. Meteorology*, **36**:401 (1979).

32. R. R. Draxler, "Determination of Atmospheric Diffusion Parameters," *Atmospheric Environment*, **10**:99 (1976).

33. R. L. Mancuso, F. L. Ludwig, "User's Manual for the APRAC-1A Urban Diffusion Model Computer Program," Stanford Research Institute, CA, 1972.

34. J. P. Nelli, A. D. Messina, J. A. Bullin, "Analysis and Modeling of Air Quality at Street Intersections," *J. Air Pollution Control Association*, **33**:760 (1983).

35. S. H. Cadle, D. P. Chock, J. M. Heuss, P. R. Monson, "Results of the General Motors Sulfate Dispersion Experiments," General Motors Research Laboratories, GMR-2107, Warren, MI, 1976.

36. J. R. Martinez, H. S. Javitz, R. E. Ruff, A. Valdes, K. C. Nitz, W. F. Dabberdt, "Methodology for Evaluating Highway Air Pollution Dispersion Models," National Cooperative Highway Research Program Rep. 245, Trans. Research Board, NRC, Washington, D. C., 1981.

37. J. A. Bullin, A. D. Messina, J. P. Nelli, "Vehicle Emissions at Intersections," Texas Transportation Institute Report 250-1F, Texas A&M University, College Station, TX, 1982.

38. W. F. Dabberdt, E. Shelar, D. Marimont, G. Skinner, "Analyses, Experimental Studies, and Evaluations of Control Measures for Air Flow and Air Quality on and Near Highways; Vol. I—Exper. Studies, Analyses, and Model Development," Federal Highway Administration Report FHWA/RD-81/051, 1981.

39. Personal Communication with Mr. Paul Benson of the California Department of Transportation, 1981.

40. K. E. Noll, "Air Quality Report: Interstate 40 Modification between Stratford Road and Peter's Creek Parkway in Forsyth County, North Carolina," Report for Harland Bartholomew and Assocs., Raleigh, N. C., 1973.

41. K. E. Noll, T. L. Miller, R. H. Rainey, "Final Report on the Air Monitoring Program to Determine the Impact of Highways on Ambient Air Quality," Dept. of Civil Eng., Univ. of Tennessee, Knoxville, TN, 1975.

42. W. A. Carpenter, G. G. Clemeña, "Supportive Data and Methods for the Evaluation of AIRPOL-4," Virginia Highway and Transportation Research Council, Report VHTRC 75-R57, Charlottesville, VA, 1975.

43. L. J. Habegger, T. D. Wolsko, J. E. Camaioni, "Dispersion Simulation Techniques for Assessing the Air Pollution Impacts on Ground Transportation Systems," Report for Argonne National Laboratory, Argonne, IL., 1974.

44. A. J. Ranzieri, G. R. Bemis, E. C. Shirley, "Air Pollution and Roadway Location, Design, and Operation," California Division of Transportation Report No. A-DOT-TL-7080-75-15, Sacramento, CA, 1975.

45. F. I. Badgely, A. T. Rossano, D. Lutrick, H. Alsid, "The Selection and Calibration of Air Quality Diffusion Models for Washington State. Highway Line Sources," Research Program Report 12.2, Project Y-1540, Washington State Department of Highways, Olympia, WA, 1976.

46. S. T. Rao, M. Chen, G. Sistla, "Dispersion of Pollutants Near Highways—Experimental Design and Data Acquisition Procedures," EPA-600/4-78-037, 1978.

47. J. H. Schroeder, "TXLINE-2: A Finite Length Model to Simulate the Dispersion of Pollutants from Roadways," M. S. Thesis, Texas A&M University, College Station, TX, 1985.

48. J. J. Korpics, "TEXIN2: A Model for Predicting Carbon Monoxide Concentrations Near Intersections," M. S. Thesis, Texas A&M University, College Station, TX, 1985.

49. "Development of an Improved Highway Capacity Manual," Final Report, National Cooperative Highway Research Program Report 3-28, Aug., 1979.

50. C. J. Messer, D. B. Fambro, "A New Critical Lane Analysis for Intersection Design," Paper for the 56th Annual Meeting of the TRB, Washington, D. C., Jan., 1977.

51. "Quick-Response Urban Travel Estimation Techniques and Transferable Parameters User's Guide," National Cooperative Highway Research Program Report 187, 1978.

52. Correspondence with A. Parikh, New Jersey State Department of Transportation, February, 1985.

53. W. R. Reilly, C. C. Gardner, J. H. Kell, "A Technique for Measurement of Delay at Intersections," FHWA Offices of Research and Development, FHWA/RD-76-135, Sept., 1976.

54. D. Ismart, "Mobile Source Emissions and Energy Analysis at an Isolated Intersection," U. S. DOT, Federal Highway Administration, Urban Planning Division, 1981.

55. "Mobile Source Emission Factors in Highway-Project Analysis," U. S. DOT, Federal Highway Administration, FHWA Technical Advisory T6640.3.

56. Personal Communication with the MOBILE3 Technical Support Staff, U. S. Environmental Protection Agency, Ann Arbor, MI, Dec., 1984.

57. J. A. Bullin, J. J. Korpics, M. W. Hlavinka, "User's Guide to the TEXIN2 Model—A Model for Predicting Carbon Monoxide Concentrations Near Intersections," Texas Transportation Institute Research Report 283-2, Texas A&M University, College Station, TX, 1986.

58. "Anti-Tampering and Anti-Misfueling Programs to Reduce In-Use Emissions from Motor Vehicles," U. S. EPA, Ann Arbor, MI, EPA-AA-TSS-83-10, 1983.

59. R. Winfrey, *Economics Analysis for Highways*, International Textbook Company, Scranton, PA (1969).

60. P. E. Benson, "CALINE3—A Versatile Dispersion Model for Predicting Air Pollution Levels Near Highways and Arterial Streets," California Department of Transportation, Sacramento, California, 1979.

61. D. P. Chock, "A Simple Line-Source Model for Dispersion Near Roadways," *Atmospheric Environment*, Vol. 12, 1978.

62. J. A. Bullin, J. C. Polasek, "Traffic Measurement for Roadway Pollution Studies by Radar Methods," *J. Air Pollution Control Association*, **28**:158 (1978).

63. J. B. Rodden, "A Non-Fickian Gradient Transport Model to Predict Air Pollution Dispersion from Roadways," M. S. Thesis, Texas A&M University, College Station, TX, 1983.

64. F. B. Smith, "A Scheme for Estimating the Vertical Dispersion of a Plume from a Source Near Ground Level," *Air Pollution—Modeling*, No. 14, CCMS/NATO, 1972.

65. Personal Communication with Mr. Paul Benson of the California Department of Transportation, June, 1986.

66. G. C. Gill, "Development and Use of the Gill UVW Anemometer," 3rd Symposium on Meteorological Observations and Instrumentation, Amer. Meteor. Soc., Boston, MA.

67. "Protection of Environment," Code of Federal Regulations, Office of the Federal Register, National Archives and Records Administration, Vol. 40, Part 50, Subchapter C, 1985.

68. J. A. Bullin, N. J. Green, J. C. Polasek, "Determination of Vehicle Emission Rates from Roadways by Mass Balance Techniques," *Environmental Science and Technology*, **14**:6 (1980).

Nomenclature

Nomenclature

$a =$ $(1+m)/(1+2m)$

$ADPV =$ approach delay (sec/veh)

ASDA = average stop delay of all vehicles on inbound approach (sec/veh)

ASDL = average stop delay of left turns on inbound approach (sec/veh)

ASDR = average stop delay of right turns on inbound approach (sec/veh)

ASDS = average stop delay of straights on inbound approach (sec/veh)

ATDA = average total delay of all vehicles on inbound approach (sec/veh)

ATDL = average total delay of left turns on inbound approach (sec/veh)

ATDR = average total delay of right turns on inbound approach (sec/veh)

ATDS = average total delay of straights on inbound approach (sec/veh)

$b =$ $2/(1+2m)$

$C =$ pollutant concentration

$CNT =$ number of reflections required for convergence

COBK = total CO emission on inbound bucket nearest intersection kg/15 min

$COID =$ excess emissions due to vehicles idling

COOP = total CO emission on outbound approach kg/15 min

$COSD =$ excess emissions due to slowing

$COST =$ total amount of excess carbon monoxide emitted due to vehicles stopping (lb/hr)

$CY =$ cycle time (sec)

$D =$ line source length

$ER =$ carbon monoxide emitted per 1000 speed changes (lb)

$EF =$ emission factor gm/m · sec

FLS = finite line source

${}_1F_1 =$ Kummer's function defined in equation (2-21)

$h =$ source height

H = height that pollutant emitted from centerline would reach by the time it reached the roadway edge

HDDV = heavy duty Diesel vehicles

HDGV = heavy duty gasoline vehicles

HRS = excess hours consumed per 1000 speed changes

HYP = cosine of the angle LB

$I_{-\frac{m}{1+2m}}$ = Bessel function of the first kind with order equal to $-m/(1+2m)$

k = 0.4 (known as von Kármán's constant)

K_i = eddy diffusivity ($i = x, y, z$) in the x, y, and z directions, respectively

K_1 = eddy diffusivity at $z = z_1$

l = length along line sources in equation (2-9)

L = atmospheric mixing height

LB = bearing of the link with respect to the x-axis

LDDT = light duty Diesel vehicles

LDDV = light duty Diesel vehicles

LDGT1 = light duty gasoline trucks with a gross vehicle weight rating (GVWR) less than 6001 lbs

LDGT2 = light duty gasoline trucks with a gross vehicle weight rating (GVWR) less than 8501 lbs

LDGV = light duty gasoline vehicles

$LLEN$ = length of link

LT = fraction of left turning vehicles

m = power law wind speed parameter from $u(z) = u_1 \left(\frac{z}{z_1}\right)^m$

MC = motorcycles

M_H = conflicting traffic streams as described in Figure 29

M_{NO} = maximum capacity for a given movement

M_1 = capacity of right streams

M_{134} = capacity of all streams using the shared lane

M_3 = capacity of through streams

M_4 = capacity of left streams

P = impedance factor defining the probability that a minor road movement will remain unaffected by traffic flow from the major road to the minor road

PCCC = percent VMT accumulated in the cold start mode by catalyst equipped vehicles

PCCN = percent VMT accumulated in the cold start mode by non-catalyst equipped vehicles

PCE = passenger car equivalency

pch = passenger cars per hour

PCHC = percent VMT accumulated in the hot start mode by catalyst equipped vehicles

$PCST$ = percent of vehicles stopping

PD = probability density function

$$PD_{ij} = \frac{1}{\sqrt{2\pi}} \int_{\frac{Y_j}{SGY_i}}^{\frac{Y_{j+1}}{SGY_i}} \exp\left(-\frac{p^2}{2}\right) dp$$

Q = pollution source strength, rate of emission

Q' = pollution source strength, rate of emission per unit length

QAVG = average queue length on approach (veh)

QE_i = central sub-element lineal source strength for ith element

QL = total queue length

QMAX = maximum queue length on approach (veh)

$SDPV$ = stopped delay per vehicle (sec)

SGY_i = σ_y as a function of downwind distance for the ith element

SGZ_i = σ_z as a function of downwind distance for the ith element

t = time

$TFLAG$ = flag indicating that TEXIN2 is to model a T-intersection

$TIQPV$ = time in queue delay (sec/veh)

TR = truck volume

$TRES$ = residence time calculated by CALINE3

$TTEI$ = total number of vehicles entering the intersection on a per lane bases (veh/hr)

u, v, w = wind velocity components in the u, v, and w directions, respectively

$\overline{u}$ = constant average wind speed in the u direction

u_{ref} = reference wind speed at z_{ref}

u_1 = reference wind speed at height $z = z_1$

u_* = friction wind velocity

VMT = vehicles miles traveled

VO = volume of traffic

$WCFLAG$ = flag indicating that TEXIN2 is to perform a worst case wind angle analysis

WT_j = source strength weighting factor for the jth FLS segment

x, y, z = directions in the Euclidean coordinate system shown in Figure 1

X, Y, Z = proportion of right, through, and left movements, respectively

XD = length in x-direction

$XLIN$ = x-coordinate of link endpoint

YD = length in y-direction

Y_j, Y_{j+1} = offset distances for the jth FLS segment

$YLIN$ = y-coordinate of link endpoint

z_{ref} = reference wind speed height

z_0 = surface roughness

z_1 = reference height taken as 1 meter

Greek Symbols

η = $\frac{u_1 z^{1+2m}}{(1+2m)^2 K_1 x}$

μ = power law constant

σ_i = standard deviation of the concentration distribution in the ith direction

Appendices

Appendix A

SETA Data Reduction Program

```
/*
*****************************************************************************
*                                                                           *
*                      SETA Data Reduction Program                          *
*                                                                           *
*                                                                           *
*                      Written by Michael W. Hlavinka                       *
*                                                                           *
*                    The Texas Transportation Institute                     *
*                                                                           *
*                    The Texas A&M University System                        *
*                                                                           *
*                             Project 22830                                 *
*                                                                           *
*                          College Station, Texas                           *
*                                                                           *
*                           in cooperation with                             *
*                                                                           *
*                          Texas State Department                           *
*                                                                           *
*                    Highways and Public Transportation                     *
*                                                                           *
*                                  and                                      *
*                                                                           *
*                    Department of Chemical Engineering                     *
*                                                                           *
*                           Texas A&M University                            *
*                                                                           *
*****************************************************************************

  This program takes the raw air quality data recorded by the balcones computer
and extracts the individual records from the raw tape blocks.  The individual
records are written to a disk file.  Preliminary checking is done on the
individual records to insure that they meet certain criteria.  If not, ap-
propriate action is taken.  All output from this program is in hexadecimal. */



#include <fcntl.h>
#include "/users/hlavinka/c/mtio.h"
#include <stdio.h>

/* The following defines ASCII backspaces, spaces, and newline */
#define  BS              0x08
#define  SP              0x20
#define  LF              0x0a
#define  STRING_LENGTH     80

struct
 {
    unsigned short int rec_length;
    unsigned short int channel;
    unsigned short rec_type;
    unsigned short int year;
    unsigned short int day;
    long unsigned int time;
 } rec_param;

struct
 {
    long unsigned int no_radar_recs;
    long unsigned int no_reg_recs;
    unsigned short int no_console_recs;
    long unsigned int radar[50];
    long unsigned int regular[50];
```

```
    unsigned short int console[50];
 } totals = {0, 0, 0};

/* Tape Control Routine. */

/* This routine allows for full software control over the tape drive */

int tapecntl (fd, control_request, how_many)
  int control_request, how_many, fd;
 {
   struct mtop tape_control;

/* Perform operation. */
    switch (control_request)
       {
         case 1:         /* Write an EOF */
           tape_control.mt_op = MTWEOF;
           tape_control.mt_count = how_many;
           break;
         case 2:         /* Forward space files. */
           tape_control.mt_op = MTFSF;
           tape_control.mt_count = how_many;
           break;
         case 3:         /* Backwards space files. */
           tape_control.mt_op = MTBSF;
           tape_control.mt_count = how_many;
           break;
         case 4:         /* Forward space records. */
           tape_control.mt_op = MTFSR;
           tape_control.mt_count = how_many;
           break;
         case 5:         /* Backwards space records. */
           tape_control.mt_op = MTBSR;
           tape_control.mt_count = how_many;
           break;
         case 6:         /* Rewind drive. */
           tape_control.mt_op = MTREW;
           tape_control.mt_count = 1;
           break;
         case 7:         /* Rewind and take offline. */
           tape_control.mt_op = MTOFFL;
           tape_control.mt_count = 1;
           break;
       }


/* Call ioctl to perform option */
    if ( control_request == 2 )
       printf ("Advancing to required file. \n");
    else if ( control_request == 6 )
       printf ("Rewinding tape drive. \n");
    if ( ioctl (fd, MTIOCTOP, &tape_control)  <0 )
      {
        printf ("System call ioctl failed. \n");
        exit (1);
      }
    close (fd);
    return (0);
 }

/* This function reads a line from the input file and removes the
   newline character from the end of the read string. */
   int read_rec (string, file_pointer, print)
     unsigned char *string;
     FILE *file_pointer, *print;
      {
        char *newline, *strchr ();
```

```
        if (fgets (string, STRING_LENGTH + 2, file_pointer) == NULL)
          fprintf (print, "fgets returned a NULL. \n");
        newline = strchr (string, '\n');
        if ( newline != NULL )
          *newline = '\0';
      }

/* This function takes two bytes and assigns each byte to the high and
   lower order bits of a two byte word. */
   unsigned short int combine (high, low)
    unsigned char high, low;
      {
        unsigned short int word;
        word = high;
        word <<= 8;
        word |= low;
        return (word);
      }


/* This function splits a two byte word into its low and high order bytes. */
   int split (word, high, low)
     unsigned short int word;
     unsigned char *high, *low;
      {
        *high = (word >> 8) & 0xff;
        *low = word & 0xff;
      }


/* This function lists the console messages to the printer file. */
   int list (record, print)
     unsigned char *record;
     FILE *print;
      {
        struct
         {
           unsigned short int hour;
           unsigned short int minute;
           unsigned short int second;
         } recording_time;
        unsigned char comment[80];
        unsigned char *com_pointer = comment, *rec_end = record + 22;
        while ( rec_end - record - 22 < 82 )
          {
            if (*rec_end == BS)
             {
               ++rec_end;
               --com_pointer;
             }
            else
             *com_pointer++ = *rec_end++;
          }

/* Calculate the time that the console message was recorded. */
        recording_time.hour = rec_param.time / 3600;
        recording_time.minute = (rec_param.time - recording_time.hour * 3600)
                / 60;
        recording_time.second = rec_param.time - recording_time.hour * 3600
                - recording_time.minute * 60;

/* List the time to the printer file. */
        fprintf (print, "%02d:%02d:%02d:  ",
           recording_time.hour, recording_time.minute, recording_time.second);

/* List the comment. */
        com_pointer = comment;
```

```
        while (*com_pointer != '\0')
          fprintf (print, "%c", *com_pointer++);
        fprintf (print, "\n");
      }


/* This function modifies the original raw data records by the insertion of
   a date stamp.  In addition, the radar recors are modified so that all
   information up to the record type is the same format as the regular
   and console records.  The final record formats are:

   Radars:

   |length|ch #|time|time|year|day|name|name|name|name|type|# cat1|spd 1|
   |spd 1|# cat2|spd 2|spd 2|# cat3|spd 3|spd 3|# cat4|spd 4|spd 4|# cat5|
   |spd 5|spd 5|

   Regular channels:

   |length|ch #|time|time|year|day|name|name|name|name|type|data|type|data|
   |type|data|type|data|type|data|type|data|

   Console messages:

   ||length|0000H|time|time|year|day|CO|MM|EN|T |type|console message------|

   Each of the entries consists of two bytes and the time stamp is spread
   over four bytes and the name over eight bytes.  The channel number
   is not applicable to console messages (0000H is inserted).  The console
   messages are padded with spaces to fill the entire record.     */

   int modify (data_pointer, record)
     unsigned char *data_pointer, *record;
     {
       int n;
       unsigned char high, low;

/* Add 4 bytes to the current record length. */
       rec_param.rec_length += 4;

/* Place modified record length in data. */
       split (rec_param.rec_length, &high, &low);
       *data_pointer = high;
       *(data_pointer + 1) = low;

/* Modify the radar records. */
       if (rec_param.rec_length == 52)
         {
           for ( n = 0; n <= 3; ++n)
             *(record + n) = *(data_pointer + n);
           for ( n = 4; n <= 7; ++n)
             *(record + n) = *(data_pointer + n + 8);

/* Convert year and day words into their low and high order bytes and place
   in record. */
           split (rec_param.year, &high, &low);
           *(record + 8) = high;
           *(record + 9) = low;
           split (rec_param.day, &high, &low);
           *(record + 10) = high;
           *(record + 11) = low;

/* Now place the channel name in record... */
           for ( n = 12; n <= 19; ++n)
              *(record + n) = *(data_pointer + n - 8);

/* and finally fill out the remainder of record with record type and data. */
```

```
            for ( n = 20; n < rec_param.rec_length; ++n )
               *(record + n) = *(data_pointer + n - 4);
          }

        else   /* Now regular and console message records. */
          {
            for ( n = 0; n <= 7; ++n )
               *(record + n) = *(data_pointer + n);

/* Split year and day into single bytes and place in record. */
            split (rec_param.year, &high, &low);
            *(record + 8) = high;
            *(record + 9) = low;
            split (rec_param.day, &high, &low);
            *(record + 10) = high;
            *(record + 11) = low;

/* Now fill record with the remaining bytes. */
            for ( n = 12; n < rec_param.rec_length; ++n )
               *(record + n) = *(data_pointer + n - 4);
          }
      }


/* This function converts the Julian dates to calendar dates. */
   int convert (year, day, print)
   unsigned short int year, day;
   FILE *print;
    {
      static struct
       {
         unsigned short int number_of_days;
         char name[3];
       } months[12] = { { 31, {'J', 'a', 'n'} }, { 28, {'F', 'e', 'b'} },
                        { 31, {'M', 'a', 'r'} }, { 30, {'A', 'p', 'r'} },
                        { 31, {'M', 'a', 'y'} }, { 30, {'J', 'u', 'n'} },
                        { 31, {'J', 'u', 'l'} }, { 31, {'A', 'u', 'g'} },
                        { 30, {'S', 'e', 'p'} }, { 31, {'O', 'c', 't'} },
                        { 30, {'N', 'o', 'v'} }, { 31, {'D', 'e', 'c'} } };
      int i, sum = 0, date;

/* Check if the given year is a leap year. */
      if ( (year % 4 == 0 && year % 100 != 0) || year % 400 == 0 )
        months[1].number_of_days = 29;    /* Is a leap year. */
      else
        months[1].number_of_days = 28;    /* Not a leap year. */

/* Determine date. */
      for ( i = 0; i < 12; ++i )
        {
          sum += months[i].number_of_days;
          if ( sum >= day )
            break;
        }
      date = day - (sum - months[i].number_of_days);
      fprintf (print, "%c%c%c. %02d, %04d\n\n", months[i].name[0],
            months[i].name[1], months[i].name[2], date, year);
    }


/* Main Program begins here...*/

main (argc, argv)
 int argc;
 char *argv[];
                        /* Command line arguments include the comment file,
```

```
                              the grouping file, number of files to process,
                              the output file name, and the printer file name. */
{
   unsigned char data[4000], record[105], *rec_end;
   unsigned char *data_pointer = data, *string_pointer;
   unsigned char comment[105], space;
   unsigned int fd_in, blksize, block = 0;
   short unsigned int no_files_processed, output_group = 0, group_count;
   short unsigned int files_to_process;
   int n, i;
   unsigned short *date_comments;
   FILE *date_comment_file, *grouping_file, *output_file, *print, *fopen ();

/* Open the date and comment file and the output grouping file. */
   date_comment_file = fopen (argv[1], "r");
   grouping_file = fopen (argv[2], "r");
   if ( fscanf (grouping_file, "%hd", &group_count) != 1 )
     fprintf (print, "fscanf did not assign the correct number of values.\n");
   output_group += group_count;

/* Now open the output files. */
   output_file = fopen (argv[4], "w");      /* This is the raw data file. */
   print = fopen (argv[5], "a");            /* This is the printer file. */
   fprintf (print, "\f\nConsole messages and recording times:\n");

/* Initialize variables. */
   no_files_processed = 0;
   space = SP;

/* Determine the number of input files to process. */
   if ( sscanf (argv[3], "%hd", &files_to_process) != 1 )
     fprintf (print, "sscanf did not assign the correct number of values.\n");
   for ( n = 0; n < files_to_process; ++n)
     {
       *(totals.radar + n) = 0;
       *(totals.regular + n) = 0;
       *(totals.console + n) = 0;
     }

/* Open tape drive for input. */
   if ( (fd_in = open ( "/dev/rmt9", O_RDONLY) ) < 0 )
     {
       printf ("Tape drive was not successfully opened. \n");
       exit (1);
     }

/* Begin processing data. */

   while ( no_files_processed < files_to_process )
     {

/* Read the Julian date the file was recorded. */
       read_rec (comment, date_comment_file, print);
       if (sscanf (comment, "%4hd.%3hd", &rec_param.year, &rec_param.day) != 2)
         fprintf (print, "sscanf did not assign the correct number of values.\n"
            );

/* Place the length at the beginning of the comment. */
       *comment = 0;              /* This and next line sets comment length. */
       *(comment + 1) = 0x68;

/* Place a date and time stamp on the record.  Also add other information
   so that this comment looks like comments generated by the data acquisition
   program. */
       for ( i = 2; i <= 7; ++i )
         *(comment + i) = 0;
       date_comments = (unsigned short *) (comment + 8);
```

```
        *date_comments = rec_param.year;
        date_comments = (unsigned short *) (comment + 10);
        *date_comments = rec_param.day;
        sprintf (comment+12, "COMMENT ");
        *(comment + 20) = 0;
        *(comment + 21) = 0x14;

/* Read the comment and write it to the output file. */
        string_pointer = comment + 22;
        read_rec (string_pointer, date_comment_file, print);

/* Pad comment with spaces if length is less than 104 bytes (82 bytes of
   ASCII comments). */
        while ( strlen (comment+22) < 82 )
          strcat (comment+22, &space);

/* Write the comment string to the output files. */
        rec_end = comment;
        while ( rec_end - comment < 104 )
            fprintf (output_file, "%02x", *rec_end++);
        fprintf (output_file, "\n");
        fprintf (print, "%s\n", comment+22);

/* Now write the date to the printer output file. */
        fprintf (print, "Date this file was recorded - %04d.%03d ---- ",
                 rec_param.year, rec_param.day);
        convert (rec_param.year, rec_param.day, print);

/* Begin processing input tape data. */
        do
          {
            ++block;
            if ( (blksize = read (fd_in, data, 4000) ) < 0 )   /* Read a block */
               {
                fprintf (print, "Error in reading block %d ", block);
                fprintf (print, "in file %d.  Processing next file.\n",
                   no_files_processed + 1);
                tapecntl (fd_in, 2, 1);
               }

/* If block was correctly read and drive is not at EOF, process data. */
            if ( blksize > 0 )
              {

                do
                  {
/* Determine the record length. */
                    rec_param.rec_length = combine (*data_pointer,
                        *(data_pointer + 1));

/* Insure that the record length is compatible with the balcones software. */
                    if (rec_param.rec_length != 40 && rec_param.rec_length != 48
                         && rec_param.rec_length != 100)
                      {
                        fprintf (print, "File %d, block %d has a record ",
                          no_files_processed + 1, block);
                        fprintf (print, "length of %d bytes.  Process ",
                          rec_param.rec_length);
                        fprintf (print, "next block.\n");
                        break;
                      }

/* Insure that the record type is valid.  If not valid, issue warning
   but continue processing data if that is the only error. */
                    rec_param.rec_type = combine (*(data_pointer + 16),
                       *(data_pointer + 17));
                    if (rec_param.rec_type < 0 || rec_param.rec_type > 20)
```

```
                    {
                      fprintf (print, "File %d, block %d has a record type ",
                         no_files_processed + 1, block);
                      fprintf (print, "of %d.  Continuing in same block.\n",
                         rec_param.rec_type);
                    }

/* Check the value of the time stamp.  It should be between 0 and 172,800
   seconds (48 hrs.).  Subsequently check the channel number range.  The
   time stamp location depends upon the type of record. */
                  if (rec_param.rec_length == 48)      /* Radar records */
                    {

/* Evaluate the time stamp for radar records. */
                      rec_param.time = combine (*(data_pointer + 12),
                        *(data_pointer + 13));
                      rec_param.time <<= 16;
                      rec_param.time |= combine (*(data_pointer + 14),
                         *(data_pointer + 15));
                      if (rec_param.time < 0 || rec_param.time > 172800)
                        {
                          fprintf (print, "File %d, block %d has a radar ",
                            no_files_processed + 1, block);
                          fprintf (print, "time stamp of %d.  Next block.\n",
                            rec_param.time);
                          break;
                        }

/* Check the channel number. */
                      rec_param.channel = combine (*(data_pointer + 2),
                          *(data_pointer + 3));
                      if (rec_param.channel < 0 || rec_param.channel > 10)
                        {
                          fprintf (print, "File %d, block %d has a radar ",
                            no_files_processed + 1, block);
                          fprintf (print, "channel of %d.  Next block.\n",
                            rec_param.channel);
                          break;
                        }
                      if (rec_param.rec_type == 10)
                        ++totals.radar[no_files_processed];
                    }
                  else   /* Regular channel and console message records. */
                    {

/* Evaluate the time stamp for regular channel and console records. */
                      rec_param.time = combine (*(data_pointer + 4),
                         *(data_pointer + 5));
                      rec_param.time <<= 16;
                      rec_param.time |= combine (*(data_pointer + 6),
                           *(data_pointer + 7));
                      if ( (rec_param.time < 0 || rec_param.time > 172800)
                          && rec_param.rec_length == 40 )
                        {
                          fprintf (print, "File %d, block %d has a %d byte ",
                               no_files_processed + 1, block,
                               rec_param.rec_length);
                          fprintf (print, "record with time stamp %d.",
                               rec_param.time);
                          fprintf (print, "  Process next block.\n");
                          break;
                        }

/* If the record is for a regular channel, check the channel number. */
                      if (rec_param.rec_length == 40)
                        {
                          rec_param.channel = combine (*(data_pointer + 2),
```

```
                                *(data_pointer + 3));
                            if (rec_param.channel < 11 || rec_param.channel > 64)
                              {
                                fprintf (print, "File %d, block %d has a ",
                                  no_files_processed + 1, block);
                                fprintf (print, "channel number of %d.  ",
                                  rec_param.channel);
                                fprintf (print, "Processing next block.\n");
                                break;
                              }
                            ++totals.regular[no_files_processed];
                          }
                        else if (rec_param.rec_length == 100)
                          ++totals.console[no_files_processed];
                      }

/* Modify the record by addition of the date stamp.  Also if the record is
   a radar record, further modify it so that the channel name and time stamp
   is like that of non-radar and console records. */
                      modify (data_pointer, record);
                      data_pointer += rec_param.rec_length - 4;
                      rec_end = record;

/* If record has made it this far, it should be correct.  List comment record
   at this point. */
                      if (rec_param.rec_type == 20)
                        {
/* Change ASCII spaces in comment record following record length to ASCII
   NULL characters. */
                          *(record + 2) = '\0';
                          *(record + 3) = '\0';
                          list (record, print);
                        }

/* Write the record to the output file. */
                      while (rec_end - record < rec_param.rec_length)
                        fprintf (output_file, "%02x", *rec_end++);
                      fprintf (output_file, "\n");
                    }
                  while (data_pointer - data < blksize);
                }
              data_pointer = data;
            }
          while ( blksize > 0 );
          fprintf (print, "\n");
          block = 0;
          ++no_files_processed;
          if (no_files_processed == output_group)
            {
              fprintf (print, "\f\nConsole messages and recording times:\n");
              if ( fscanf (grouping_file, "%hd", &group_count) != 1 )
                {
                  fprintf (print, "fscanf did not assign the correct number of ");
                  fprintf (print, "values.\n");
                }
              output_group += group_count;
            }
        }

/* Print the report on the total number of all types of records. */
      for ( n = 0; n < files_to_process; ++n )
        {
          fprintf (print, "Radar records in file %d = %d\n", n + 1,
              *(totals.radar + n));
          fprintf (print, "Regular records in file %d = %d\n", n + 1,
              *(totals.regular + n));
          fprintf (print, "Console records in file %d = %d\n", n + 1,
```

```
            *(totals.console + n));
        fprintf (print, "\n");
        totals.no_radar_recs += *(totals.radar + n);
        totals.no_reg_recs += *(totals.regular + n);
        totals.no_console_recs += *(totals.console + n);
      }
    fprintf (print, "\n");
    fprintf (print, "Total record counts on input tape:\n");
    fprintf (print, "Total radar records = %d\n", totals.no_radar_recs);
    fprintf (print, "Total regular records = %d\n", totals.no_reg_recs);
    fprintf (print, "Total console message records = %d\n",
        totals.no_console_recs);
    tapecntl (fd_in, 7, 1);

/* Close all files. */
    close (fd_in);
    fclose (output_file);
    fclose (print);
    fclose (grouping_file);
  }
```

Appendix B

SETB Data Reduction Program

```
/*
*****************************************************************************
*                                                                           *
*                      SETB Data Reduction Program                          *
*                                                                           *
*                                                                           *
*                     Written by Michael W. Hlavinka                        *
*                                                                           *
*                   The Texas Transportation Institute                      *
*                                                                           *
*                    The Texas A&M University System                        *
*                                                                           *
*                            Project 22830                                  *
*                                                                           *
*                        College Station, Texas                             *
*                                                                           *
*                          in cooperation with                              *
*                                                                           *
*                        Texas State Department                             *
*                                                                           *
*                   Highways and Public Transportation                      *
*                                                                           *
*                                 and                                       *
*                                                                           *
*                   Department of Chemical Engineering                      *
*                                                                           *
*                          Texas A&M University                             *
*                                                                           *
*****************************************************************************

 The purpose of this program is to take sorted data from the SETA program
 and calculate calibration data to be used in the SETC program.  The file
 that contains the SETA data is given on the command line.                 */


#include <stdio.h>

struct
   {
      unsigned short int rec_length;
      unsigned short int channel;
      short int type[6];
      short int sample[6];
      unsigned short int year;
      unsigned short int day;
      unsigned long int time;
   } rec_param;

enum flag {false, true};

main (argc, argv)
 int argc;
 char *argv[];
 {
   enum flag flush;
   unsigned short int no_cal_channels, n = 0, key_channel;
   short int current_type = 0, no_samples = 0;
   long int average = 0;
   unsigned short int cal_channels[25];
   short int *sample_pt = rec_param.sample, *type_pt = rec_param.type;
   unsigned char record[220], filename[14];
   FILE *infile, *fopen (), *cal_file[25], *sort;

/* Read the number of calibration channels and the channel numbers. */
   if ( (infile = fopen (argv[1], "r") ) == NULL )
     fprintf (stderr, "Infile could not be opened.\n");
   fscanf (infile, "%hd", &no_cal_channels);
```

```
    while ( fscanf (infile, "%hd", cal_channels + n) != EOF )
      ++n;
    n = 0;

/* Open the files for the calibration channels--1 file for each cal channel */
    for ( n = 0; n < no_cal_channels; ++n )
        {
          sprintf (filename, "channel%2d.cal", *(cal_channels + n) );
          if ( (cal_file[n] = fopen (filename, "w") ) == NULL )
            {
              fprintf (stderr, "Could not create temporary file.\n");
              exit (1);
            }
        }
    n = 0;

/* Open sorted data file. */
    if ( (sort = fopen (argv[2], "r") ) == NULL )
      {
        fprintf (stderr, "Sorted file could not be opened.\n");
        exit (1);
      }
    key_channel = *(cal_channels + n);
    average = 0;

/* Process the sorted data.  Calculate the average values for calibration
    data and place them in their individual temporary files. */
    while ( fgets (record, 220, sort) != NULL )
      {
        sscanf (record + 4, "%4hx", &rec_param.channel);
        if ( rec_param.channel > key_channel && n != (no_cal_channels - 1) )
          {
            if ( flush )    /* If average data needs dumping do it. */
              {
                average /= no_samples;
                fprintf (cal_file[n], "%.4hx%.4hx\n", current_type,
                      (short)average );
                no_samples = 0;
                average = 0;
                current_type = 0;
                flush = false;
                fflush (cal_file[n]);
              }
            ++n;
            key_channel = *(cal_channels + n);
          }
        else if ( rec_param.channel < key_channel && n == (no_cal_channels - 1) )
          {
            if ( flush )    /* If average data needs dumping do it. */
              {
                average /= no_samples;
                fprintf (cal_file[n], "%.4hx%.4hx\n", current_type,
                    (short)average );
                no_samples = 0;
                average = 0;
                flush = false;
                current_type = 0;
                fflush (cal_file[n]);
              }
            if ( rec_param.channel < *(cal_channels + n - 1) )
              {
                n = 0;
                key_channel = *cal_channels;
              }
          }
        if ( rec_param.channel == key_channel )   /* Correct channel. Process. */
          {
```

```
/* Get the record types and samples from the record. */
          sscanf (record + 40, "%4hx%4hx%4hx%4hx%4hx%4hx%4hx%4hx%4hx%4hx%4hx\
%4hx", rec_param.type, rec_param.sample, rec_param.type+1, rec_param.sample+1,
rec_param.type+2, rec_param.sample+2, rec_param.type+3, rec_param.sample+3,
rec_param.type+4, rec_param.sample+4, rec_param.type+5, rec_param.sample+5 );

/* If any of the record types are cal data process the data. */
          type_pt = rec_param.type;
          sample_pt = rec_param.sample;
          for (   ; (type_pt - rec_param.type) < 6; ++type_pt, ++sample_pt )
              {
                if ( *type_pt == 1 || *type_pt == 2 )
                  {
                    if ( no_samples == 0 )
                      {
                        fprintf (cal_file[n], "%.36s", record + 4);
                        current_type = *type_pt;
                      }
                  }

/* Dump average to output correct output file. */
                if ( current_type != *type_pt && *type_pt != 0
                        && current_type != 0 )
                  {
                    flush = false;
                    average /= no_samples;
                    fprintf (cal_file[n], "%.4hx%.4hx\n", current_type,
                             (short)average );
                    no_samples = 0;
                    average = 0;
                    current_type = 0;
                    --type_pt;
                    --sample_pt;
                    fflush (cal_file[n]);
                  }

/* Sum cal data to obtain an average. */
                if ( *type_pt == current_type && (current_type == 1
                          || current_type == 2) )
                  {
                    flush = true;
                    average += *sample_pt;
                    ++no_samples;
                  }
                if ( *type_pt == 99 || *type_pt == -1 )
                  {
                    no_samples = 0;
                    fprintf (cal_file[n], "%.36s%.4hx0000\n", record + 4,
                           *type_pt);
                  }
              }
          }
      }
  if ( flush )
    {
      average /= no_samples;
      fprintf (cal_file[n], "%.4hx%.4hx\n", current_type, (short)average );
      fflush (cal_file[n]);
    }
  for ( n = 0; n < no_cal_channels; ++n )
    fclose (cal_file[n]);
}
```

```
/*  The purpose of this program is to calculate calibration drift factors
    for all of the calibration channels:  The output of this program is to
    be used in the setc program to correct for calibration and zero drift.
    Input to the program consists of the averaged calibration data from setb.
    This data is sorted by the following keys:  channel, year, day, and time.
    The resulting data from this program is again sorted in the following
    manner:  year, day, time, channel.  This program is used only on channels
    that contain BOTH a zero and span drift.  A condensed version is used on
    those instruments that only contain span drifts (e.g., Byron 401).  */

#include <stdio.h>

FILE *input, *output, *fopen ();
unsigned char record[48];

new_data (zero_st, span_st, time_st, day_st, channel)
 long int *time_st;
 short int *zero_st, *span_st, *day_st;
 short int channel;
 {
  short int read_channel, type, sample;
  if (fgets (record, 48, input) == NULL)
    exit (1);
  sscanf (record, "%4hx", &read_channel);
  if (read_channel != channel)
    exit (1);
  sscanf (record + 4, "%8lx", time_st);
  sscanf (record + 16, "%4hx", day_st);
  sscanf (record + 36, "%4hx", &type);
  sscanf (record + 40, "%4hx", &sample);
  if (type == 1)
    *zero_st = sample;
  else if (type == 2)
    *span_st = sample;
  if (fgets (record, 48, input) == NULL)
    exit (1);
  sscanf (record, "%4hx", &read_channel);
  if (read_channel != channel)
    exit (1);
  sscanf (record + 36, "%4hx", &type);
  sscanf (record + 40, "%4hx", &sample);
  if (type == 1)
    *zero_st = sample;
  else if (type == 2)
    *span_st = sample;
 }

main (argc, argv)
 int argc;
 char *argv[];
 {
   unsigned char out_rec[48];
   short int span_st, span_end, zero_st, zero_end, day_st, day_end;
   long int time_st, time_end, time_diff;
   short int channel, read_chan, year, first_type, sample, sec_type;
   if ( (input = fopen (argv[1], "r") ) == NULL )
     {
       fprintf (stderr, "Could not open input file.\n");
       exit (1);
     }
   if ( (output = fopen (argv[2], "a") ) == NULL )
     {
       fprintf (stderr, "Could not open output file.\n");
       exit (1);
     }
   sscanf (argv[3], "%hd", &channel);
   fprintf (stderr, "Processing channel %d\n", channel);
   while (fgets (record, 48, input) != NULL)
```

```
      {
        sscanf (record, "%4hx", &read_chan);
        if (read_chan == channel)
          {
            sscanf (record + 12, "%4hx", &year);
            sscanf (record + 36, "%4hx", &first_type);
            fseek (input, -45, 1);
            if (first_type != -1)
              new_data (&zero_st, &span_st, &time_st, &day_st, channel);
            break;
          }
      }
    while (fgets (record, 48, input) != NULL)
      {
        sscanf (record, "%4hx", &read_chan);
        if (read_chan != channel)
          break;
        sscanf (record + 4, "%8lx", &time_end);
        sscanf (record + 16, "%4hx", &day_end);
        sscanf (record + 36, "%4hx", &first_type);
        sscanf (record + 40, "%4hx", &sample);
        if (first_type == 1)
          zero_end = sample;
        else if (first_type == 2)
          span_end = sample;
        else if (first_type == 99)
          {
            span_end = span_st;
            zero_end = zero_st;
            fprintf (output, "%.4hx%.8lx%.4hx%.4hx%.4hx%.8lx%.4hx%.4hx%.4hx%.4hx\
 \n", channel, time_st, year, day_st, day_end, time_end, span_st, span_end,
 zero_st, zero_end);
            time_st = time_end + 1;
          }
        else if (first_type == -1)
          new_data (&zero_st, &span_st, &time_st, &day_st, channel);
        if (first_type != -1 && first_type != 99)
          {
            fgets (record, 48, input);
            sscanf (record + 36, "%4hx", &sec_type);
            sscanf (record + 40, "%4hx", &sample);
            if (sec_type == 1)
              zero_end = sample;
            else if (sec_type == 2)
              span_end = sample;
            else if (sec_type == 99)
              {
                span_end = span_st;
                zero_end = zero_st;
                fprintf (output, "%.4hx%.8lx%.4hx%.4hx%.4hx%.8lx%.4hx%.4hx%.4hx%\
.4hx\n", channel, time_st, year, day_st, day_end, time_end, span_st, span_end,
zero_st, zero_end);
                time_st = time_end + 1;
              }
            else if (sec_type == -1)
              new_data (&zero_st, &span_st, &time_st, &day_st, channel);
            if ((first_type == 1 || first_type == 2) && (sec_type == 1 ||
                      sec_type == 2))
              {
                fprintf (output, "%04hx%08lx%04hx%04hx%04hx%08lx%04hx%04hx%04hx%\
04hx\n", channel, time_st, year, day_st, day_end, time_end, span_st, span_end,
zero_st, zero_end);
                time_st = time_end + 1;
              }
          }
      }
  }
```

```
/*  The purpose of this program is to calculate calibration drift factors
    for all of the calibration channels.  The output of this program is to
    be used in the setc program to correct for calibration and zero drift.
    Input to the program consists of the averaged calibration data from setb.
    This data is sorted by the following keys:  channel, year, day, and time.
    The resulting data from this program is again sorted in the following
    manner:  year, day, time, channel.  This program is used only on channels
    that contain ONLY a span drift.                                          */

#include <stdio.h>

FILE *input, *output, *fopen ();
unsigned char record[48];

new_data (span_st, time_st, day_st, channel)
 long int *time_st;
 short int *span_st, *day_st;
 short int channel;
 {
  short int read_channel, type, sample;
  if (fgets (record, 48, input) == NULL)
    exit (1);
  sscanf (record, "%4hx", &read_channel);
  if (read_channel != channel)
    exit (1);
  sscanf (record + 4, "%8lx", time_st);
  sscanf (record + 16, "%4hx", day_st);
  sscanf (record + 36, "%4hx", &type);
  sscanf (record + 40, "%4hx", &sample);
  if (type == 2)
    *span_st = sample;
 }

main (argc, argv)
 int argc;
 char *argv[];
 {
   unsigned char out_rec[48];
   short int span_st, span_end, zero_st, zero_end, day_st, day_end;
   long int time_st, time_end, time_diff;
   short int channel, read_chan, year, first_type, sample;
   if ( (input = fopen (argv[1], "r") ) == NULL )
     {
       fprintf (stderr, "Could not open input file.\n");
       exit (1);
     }
   if ( (output = fopen (argv[2], "a") ) == NULL )
     {
       fprintf (stderr, "Could not open output file.\n");
       exit (1);
     }
   sscanf (argv[3], "%hd", &channel);
   fprintf (stderr, "Processing channel %d\n", channel);
   while (fgets (record, 48, input) != NULL)
     {
       sscanf (record, "%4hx", &read_chan);
       if (read_chan == channel)
         {
            sscanf (record + 12, "%4hx", &year);
            sscanf (record + 36, "%4hx", &first_type);
            fseek (input, -45, 1);
            if (first_type != -1)
              new_data (&span_st, &time_st, &day_st, channel);
            break;
         }
     }
     zero_end = 0;
```

```
    zero_st = 0;
  while (fgets (record, 48, input) != NULL)
    {
      sscanf (record, "%4hx", &read_chan);
      if (read_chan != channel)
        break;
      sscanf (record + 4, "%8lx", &time_end);
      sscanf (record + 16, "%4hx", &day_end);
      sscanf (record + 36, "%4hx", &first_type);
      sscanf (record + 40, "%4hx", &sample);
      if (first_type == 2)
        {
          span_end = sample;
          zero_end = 0;
          zero_st = 0;
          fprintf (output, "%.4hx%.8lx%.4hx%.4hx%.4hx%.8lx%.4hx%.4hx%.4hx%.4hx\
\n", channel, time_st, year, day_st, day_end, time_end, span_st, span_end,
zero_st, zero_end);
          time_st = time_end + 1;
        }
      else if (first_type == 99)
        {
          span_end = span_st;
          zero_end = zero_st;
          fprintf (output, "%.4hx%.8lx%.4hx%.4hx%.4hx%.8lx%.4hx%.4hx%.4hx%.4hx\
\n", channel, time_st, year, day_st, day_end, time_end, span_st, span_end,
zero_st, zero_end);
          time_st = time_end + 1;
        }
      else if (first_type == -1)
        new_data (&span_st, &time_st, &day_st, channel);
    }
}
```

Appendix C

SETC Data Reduction Program

```
C
C
C ***********************************************************************
C *                                                                     *
C *                                                                     *
C *                                                                     *
C *                    SETC DATA REDUCTION PROGRAM                      *
C *                                                                     *
C *                                                                     *
C *                  Written by Michael W. Hlavinka                     *
C *                                                                     *
C *                 The Texas Transportation Institute                  *
C *                                                                     *
C *                  The Texas A&M University System                    *
C *                                                                     *
C *                          Project 22830                              *
C *                                                                     *
C *                      College Station, Texas                         *
C *                                                                     *
C *                       in cooperation with                           *
C *                                                                     *
C *                      Texas State Department                         *
C *                                                                     *
C *                 Highways and Public Transportation                  *
C *                                                                     *
C *                              and                                    *
C *                                                                     *
C *                 Department of Chemical Engineering                  *
C *                                                                     *
C *                       Texas A&M University                          *
C *                                                                     *
C *                                                                     *
C ***********************************************************************
C
C
      IMPLICIT INTEGER*4 (A-Z)
      CHARACTER*12 NAME(64)
      CHARACTER*8 UNITS(64)
      CHARACTER*20 LOCAL(64)
      REAL*4 CALCON(64),OFFSET(64),CCON(3),VECTOR(6,51),UVWDAT(7200,3)
      REAL*4 STDVEC(6,51),RATIO,LARGE(21),SMALL(21),AZM,ELEV
      REAL*4 CHANGE(10,6),HRMAX(64,2)
      INTEGER*2 RTYPE(64),MAXVAL(64),DATE
      INTEGER*2 MINVAL(64),INSGRP(64),RADTYP,REMAIN(54,7)
      INTEGER*4 ACTIVE(64),STATUS(64),SPDSUM(5),NOSAMP(64,12),UVW(6,3)
      INTEGER*4 FSADCT(64)
      REAL*4 AVG(54,17),STDEV(54,17),DAYMAX(64,2),RADAVG(10,5,34)
      REAL*4 ANGCOS,ANGSIN,WSPACE(34),SUM,SUM1,SUM1SQ,SUMSQ
      INTEGER*4 SRATE(64),CALDAT(64,9),START(6,3),OLDSTR(6,3),UVWCNT(12)
      INTEGER*2 HEAD(10),RADAR(16),REG(12),TYPE(6),SAMPLE(6),NOCARS(5)
      EQUIVALENCE (HEAD(1),STAMP),(RADTYP,RADAR(1)),(RADAR(2),NOCARS)
      EQUIVALENCE (RADAR(7),SPDSUM),(REG(1),TYPE),(REG(7),SAMPLE)
      EQUIVALENCE (HEAD(5),DATE)
      LOGICAL CALIB(64),PRT,NEXT,UVWFLG(12)
      COMMON RTYPE,UNITS,NAME
      COMMON /RADINF/ MONTH,DAYOFM,SHYEAR
      COMMON /COR/ LARGE,SMALL
      COMMON /UVWS/ VECTOR,STDVEC,NOUVW,UVW
      OPEN (5,FILE='/users/hlavinka/balcones/sys.dat',STATUS='OLD')
    1 FORMAT('1')
C
C Read the number of active channels.
C
      READ (5,2) NOCHAN
    2 FORMAT(I2)
C
C Read the channel parameters as given to the BALCONES computer in
```

```
C its software.
C
      READ (5,3) (ACTIVE(I),NAME(I),UNITS(I),LOCAL(I),CALCON(I),
     *OFFSET(I),RTYPE(I),MAXVAL(I),MINVAL(I),INSGRP(I),I=1,NOCHAN)
   3  FORMAT(I2,1X,A12,1X,A8,1X,A20,1X,F6.2,1X,F6.2,1X,I2,1X,I5,1X,I5,1X
     *,I3)
C
C Write the information given to the BALCONES on paper.
C
      WRITE (6,4)
      WRITE (6,5)
      WRITE (6,6) (ACTIVE(I),NAME(I),UNITS(I),LOCAL(I),CALCON(I),
     *OFFSET(I),RTYPE(I),MAXVAL(I),MINVAL(I),I=1,NOCHAN)
C
C Print the instrument grouping information that will be used in
C subsequent data reduction software.
C
      WRITE (6,7)
      WRITE (6,8) (ACTIVE(I),NAME(I),UNITS(I),LOCAL(I),INSGRP(I),I=1,
     *NOCHAN)
      WRITE (6,9)
   4  FORMAT(' ',T56,'System Configuration'/'+',54X,20('_'),//)
   5  FORMAT(' ',4X,'Channel',6X,'Name',7X,'Units',5X,'Instrument Locati
     *on',2X,'Calibration Constant',3X,'Offset',2X,'Record Type',2X,'Max
     *imum Value',2X,'Minimum Value'/'+',3X,7('_'),6X,4('_'),7X,5('_'),5
     *X,19('_'),2X,20('_'),3X,6('_'),2X,11('_'),2X,13('_'),2X,13('_'))
   6  FORMAT(' ',5X,I2,05X,A12,02X,A8,02X,A20,08X,F7.2,09X,F7.2,07X,I2,0
     *10X,I5,010X,I5)
   7  FORMAT('1',T57,'Instrument Grouping'/'+',55X,19('_'),//,T20,'Chann
     *el',13X,'Name',16X,'Units',11X,'Location',22X,'Group'/'+',19X,7('_
     *'),13X,4('_'),16X,5('_'),11X,8('_'),22X,5('_'))
   8  FORMAT(' ',021X,I2,12X,A12,11X,A8,09X,A20,011X,I3)
   9  FORMAT('0',T30,'Grouping Code:'/'+',28X,13('_'),/,T30,'First digit
     *--inst. bearing (1=North,2=East,3=South,4=West)',/,T30,'Second and
     * third digits represent height in feet',/,T30,'EXAMPLE:  330 is an
     * instrument on the south side of the road @ 30 feet.')
C
C Read how the UVW anemometers are grouped.
C
      READ (5,2) NOUVW
      READ (5,600) ((UVW(J,K),K=1,3),J=1,NOUVW)
 600  FORMAT(3I2)
C
C Read the number of days of data.
C
      READ (5,2) NODAYS
C
C Read the calibration channels and the A/D Counts for full scale on
C each calibration channel (FSADCT).
C
      DO 10 J=1,64
      CALIB(J)=.FALSE.
  10  CONTINUE
      DO 20 J=1,64
      READ (5,15,END=30) CHAN,ADVAL
  15  FORMAT (I2,1X,I4)
      CALIB(CHAN)=.TRUE.
      FSADCT(CHAN)=ADVAL
  20  CONTINUE
  30  CONTINUE
      CLOSE (5)
C
C Open calibration data file.
C
      OPEN (1,FILE='/users/hlavinka/balcones/setaout/calib.dat',
     *STATUS='OLD')
C
```

```
C Write initial headings on output.
C
      WRITE (6,1)
      WRITE (6,40)
  40  FORMAT(' ',51X,'SETC Averaging Program Output'/'+',51X,29('_'),//,
     *T20,'Agency:  Texas Transportation Institute',T70,'Site Location:
     * IH610 between N. Main and Airline Dr.',/,T30,'Project 22830',T87,
     *'Houston, Texas',/,T30,'Texas A & M University System',/,T30,'Coll
     *ege Station, Texas   77843',T71,'Collection Period:  Nov. 15 thru
     *Dec. 19, 1984',/,T30,'Texas State Department of',/,T30,'Highways a
     *nd Public Transportation',/,T30,'File D8-E',/,T30,'Austin, Texas
     *  78701',///)
      WRITE (6,50) NODAYS
  50  FORMAT(' ',T20,'Number of Collection days:  ',I2,///)
      WRITE (6,60)
  60  FORMAT(' ',T20,'Pollutants monitored:  Carbon Monixide',/,T44,'Tot
     *al Hydrocarbons',/,T44,'Non-Methane Hydrocarbons',/,T44,'Methane',
     */,T44,'Carbon Dioxide',/,T44,'Ozone',/,T44,'Nitrous Oxides (NOx)',
     */,T44,'Nitric Oxide (NO)',/,T44,'Nitrogen Dioxide',///,T20,'Traffi
     *c Monitoring',T80,'Meteorological Data'/'+',18X,19('_'),41X,19('_'
     *),/,T20,'10 Stephenson MARK V Radars',T80,'8 Wind Speed & Directio
     *n',/,T20,'5 Westbound Radars and',T80,'4 Thermometers',/,T20,'5 Ea
     *stbound Radar Units',T80,'2 Psychrometers',/,T80,'1 Barometer',/,T
     *80,'2 Pyranometers',///)
      WRITE (6,70)
  70  FORMAT(' ',T20,'Calibration Channels:  '/'+',19X,20('_'))
      DO 90 J=1,64
      IF (CALIB(J)) WRITE (6,80) J
  80  FORMAT (' ',23X,I2.2)
  90  CONTINUE
      WRITE (6,95)
  95  FORMAT(' ',//,15X,'NOTE:  UVW anemometer vector sums are a 3-D vec
     *tor sum of all components.  The bearing is the',/,23X,'azimuth ang
     *le and hence is in degrees azimuth.  The elevation angle is positi
     *ve for up',/,23X,'drafts and negative for down drafts.  It ranges
     *from +90 degrees to -90 degrees with a',/,23X,'horizontal wind hav
     *ing 0 degrees.',/)
C
C Read the elements of CHANGE.  This information marks changes in
C calibration constants that may have occurred during run due to swapping
C instruments.
C
      OPEN (UNIT=2,FILE='/users/hlavinka/balcones/changes.dat',STATUS='O
     *LD')
      READ (2,55,END=65) ((CHANGE(I,J),J=1,6),I=1,10)
  55  FORMAT(F4.0,01X,F3.0,01X,F6.0,01X,F2.0,01X,F6.2,01X,F6.2)
  65  CLOSE (2)
      SYSCON=1
C
C Begin averaging all channels.
C
      DO 165 J=1,64
      DO 175 K=1,9
      CALDAT(J,K)=0
 175  CONTINUE
 165  CONTINUE
      CALL opener (1)
C
      DO 100 KK=1,NODAYS
C
C Split files into daily temporary files.  (C program)
C
      CALL split (1,EOF)
C
C Print date of collection period.
C
      CALL print_date (YEAR, DAY, MONTH, DAYOFM, SHYEAR)
```

```
C
C Zero out STATUS.  (STATUS determines position in the temporary files.)
C
      DO 125 J=1,64
      STATUS(J)=0
 125  CONTINUE
      TIME=0
      DO 145 J=1,64
      DO 146 L=1,2
      DAYMAX(J,L)=0.
      HRMAX(J,L)=0.
 146  CONTINUE
 145  CONTINUE
C
C Zero out REMAIN.  This sets the number of samples that are left over
C from the previous record.
C
      DO 147 J=1,54
      REMAIN(J,1)=0
 147  CONTINUE
      DO 151 L=1,NOUVW
      DO 152 J=1,3
      OLDSTR(L,J)=0
      START(L,J)=0
 152  CONTINUE
 151  CONTINUE
C
C ***************
C The C must be removed if radar channels are only to be averaged.
C
C     NOCHAN=10
C
C ****************
C
C Begin Averaging.  NOTE:  Time is in seconds since midnight.
C
 160  DO 140 I=1,NOCHAN
      IF (I .EQ. 1) THEN
C
C Determine sample rates.
C
      CALL sample_rate (SRATE)
      PRT=.FALSE.
      PERIOD=0
      BEGIN=TIME
C
C Print console messages.
C
 130  CALL pass (0,TIME+3600,STAT,HEAD,RADAR,REG)
      IF (STAT .EQ. 1) GO TO 130
      END IF
      TIME=BEGIN
C
C If a change in the system configuration has occurred, perform the
C required change.
C
      IF (YEAR.GE.IFIX(CHANGE(SYSCON,1)) .AND. DAY.GE.IFIX(CHANGE(SYSCON
     *,2)) .AND. TIME .GE. IFIX(CHANGE(SYSCON,3)) .AND. ACTIVE(I).EQ.IFI
     *X(CHANGE(SYSCON,4))) THEN
      WRITE (6,105)
 105  FORMAT (' ',//,' ',132('*'),///)
      WRITE (6,115) YEAR,DAY,TIME,ACTIVE(I)
 115  FORMAT (' ',10X,'******************** CHANGE IN SYSTEM CONFIGURATI
     *ON ********************',/,15X,'Date =',I4.4,'.',I3.3,5X,'Time =',
     *I6,' seconds after midnight',5X,'Channel affected = ',I2.2,//)
      WRITE (6,85) CALCON(I),OFFSET(I),CHANGE(SYSCON,5),CHANGE(SYSCON,6)
  85  FORMAT (' ',10X,'New Configuration for channel:',/,15X,'Old calibr
```

```
     *ation cnst =',F6.2,5X,'Old offset =',F6.2,5X,'New calibration cnst
     * =',F6.2,5X,'New offset =',F6.2,/)
      WRITE (6,105)
      CALCON(I)=CHANGE(SYSCON,5)
      OFFSET(I)=CHANGE(SYSCON,6)
      SYSCON=SYSCON+1
      END IF
C
C Process an hour's data.
C
      DO 200 INT5=1,12
      IF (SRATE(ACTIVE(I)) .EQ. 0) THEN
      AVG(ACTIVE(I)-10,INT5)=10.**20
      STDEV(ACTIVE(I)-10,INT5)=10.**20
      IF (MOD(INT5,3) .EQ. 0) THEN
      AVG(ACTIVE(I)-10,(INT5-1)/3+13)=10.**20
      STDEV(ACTIVE(I)-10,(INT5-1)/3+13)=10.**20
      END IF
      IF (INT5 .EQ. 12) THEN
      AVG(ACTIVE(I)-10,17)=10.**20
      STDEV(ACTIVE(I)-10,17)=10.**20
      END IF
      GO TO 200
      END IF
      IF (INT5 .EQ. 1) THEN
C
C Initialize data areas.
C
      IF (RTYPE(I) .NE. 19) THEN
      DO 131 K=1,12
      NOSAMP(ACTIVE(I),K)=0
 131  CONTINUE
      END IF
      IF (ACTIVE(I) .LE. 10) THEN
      DO 134 K=1,5
      DO 135 L=1,34
      RADAVG(ACTIVE(I),K,L)=0.
 135  CONTINUE
 134  CONTINUE
      END IF
      IF (ACTIVE(I) .GE. 11) THEN
      DO 136 K=1,34
      WSPACE(K)=0.
 136  CONTINUE
      END IF
      IF (ACTIVE(I).GE.11 .AND. RTYPE(I).NE.19) THEN
      DO 137 K=1,17
      AVG(ACTIVE(I)-10,K)=0.
      STDEV(ACTIVE(I)-10,K)=0.
 137  CONTINUE
      END IF
      END IF
C
C The following section is for all instruments except UVW anemometers.
C
      IF (RTYPE(I) .NE. 19) THEN
      PERIOD=MAX(INT5,PERIOD)
      TIME=TIME+300
C
C If there are samples remaining from the previous record, process them
C first.  Otherwise read a record and process it.
C
      IF (REMAIN(ACTIVE(I)-10,1).NE.0 .AND. ACTIVE(I).GE.11) THEN
      SAMPNO=REMAIN(ACTIVE(I)-10,1)
      DO 430 K=1,SAMPNO
      SAMPLE(K)=REMAIN(ACTIVE(I)-10,7-SAMPNO+K)
      TYPE(K)=RTYPE(I)
```

```
 430  CONTINUE
      STATUS(ACTIVE(I))=1
      STAMP=TIME-300
      ELSE
      SAMPNO=6
      CALL pass (ACTIVE(I),TIME,STATUS(ACTIVE(I)),HEAD,RADAR,REG)
      END IF
 185  IF (STATUS(ACTIVE(I)) .EQ. 1) THEN
C
C If channel is a calibration channel, apply drift factors.
C
      IF (CALIB(ACTIVE(I)) .AND. REMAIN(ACTIVE(I)-10,1).EQ.0) THEN
      DO 500 K=1,6
      IF (((IFIX((K-1)*SRATE(ACTIVE(I))/6.+STAMP).GT.CALDAT(ACTIVE(I),4)
     * .AND. DATE.EQ.CALDAT(ACTIVE(I),3)) .OR. DATE.GT.CALDAT(ACTIVE(I),
     *3)) .AND. TYPE(K) .EQ. RTYPE(I)) THEN
 510  READ (1,520,END=530) CHAN,TIME1,YEAR,DAY1,DAY2,TIME2,SPAN1,SPAN2,Z
     *ERO1,ZERO2
 520  FORMAT(Z4,Z8,Z4,Z4,Z4,Z8,Z4,Z4,Z4,Z4)
      DTIME=(DAY2-DAY1-1)*86400+(86400-TIME1)+TIME2
      CALDAT(CHAN,1)=DAY1
      CALDAT(CHAN,2)=TIME1
      CALDAT(CHAN,3)=DAY2
      CALDAT(CHAN,4)=TIME2
      CALDAT(CHAN,5)=DTIME
      CALDAT(CHAN,6)=SPAN1
      CALDAT(CHAN,7)=SPAN2
      CALDAT(CHAN,8)=ZERO1
      CALDAT(CHAN,9)=ZERO2
      IF (CHAN .NE. ACTIVE(I)) GO TO 510
      END IF
 530  IF (TYPE(K) .EQ. RTYPE(I)) THEN
      ZDRIFT=FLOAT(CALDAT(ACTIVE(I),9)-CALDAT(ACTIVE(I),8))/FLOAT(CALDAT
     *(ACTIVE(I),5))*((K-1)*SRATE(ACTIVE(I))/6.+STAMP-CALDAT(ACTIVE(I),2
     *))
      SDRIFT=FLOAT(CALDAT(ACTIVE(I),7)-CALDAT(ACTIVE(I),6))/FLOAT(CALDAT
     *(ACTIVE(I),5))*((K-1)*SRATE(ACTIVE(I))/6.+STAMP-CALDAT(ACTIVE(I),2
     *))+CALDAT(ACTIVE(I),6)-ZDRIFT
      SAMPLE(K)=(SAMPLE(K)-ZDRIFT)*(1.+FLOAT(CALDAT(ACTIVE(I),6)-SDRIFT)
     */FLOAT(FSADCT(ACTIVE(I))))
      END IF
 500  CONTINUE
      END IF
      IF (ACTIVE(I).GE.11) REMAIN(ACTIVE(I)-10,1)=0
C
C Sum radar channel data.
C
      IF (ACTIVE(I).GE.1 .AND. ACTIVE(I).LE.10 .AND. RADTYP.NE.0) THEN
      NOSAMP(ACTIVE(I),INT5)=NOSAMP(ACTIVE(I),INT5)+1
      DO 150 K=1,5
      RADAVG(ACTIVE(I),K,INT5*2-1)=RADAVG(ACTIVE(I),K,INT5*2-1)+NOCARS(K
     *)
      RADAVG(ACTIVE(I),K,INT5*2)=RADAVG(ACTIVE(I),K,INT5*2)+SPDSUM(K)
C
C Sum data for 15 min averages.
C
      RADAVG(ACTIVE(I),K,2*((INT5-1)/3)+25)=RADAVG(ACTIVE(I),K,2*((INT5-
     *1)/3)+25)+NOCARS(K)
      RADAVG(ACTIVE(I),K,2*((INT5-1)/3)+26)=RADAVG(ACTIVE(I),K,2*((INT5-
     *1)/3)+26)+SPDSUM(K)
      RADAVG(ACTIVE(I),K,33)=RADAVG(ACTIVE(I),K,33)+NOCARS(K)
      RADAVG(ACTIVE(I),K,34)=RADAVG(ACTIVE(I),K,34)+SPDSUM(K)
 150  CONTINUE
      END IF
C
C Process data for the wind vanes.  (Type 13)
C
```

```
      IF (RTYPE(I) .EQ. 13) THEN
      INDEX1=ACTIVE(I)-10
      DO 300 K=1,SAMPNO
      IF (IFIX((K-1)*SRATE(ACTIVE(I))/6.+STAMP).GT.TIME .AND. TYPE(K).EQ
     *.RTYPE(I)) THEN
      REMAIN(INDEX1,1)=REMAIN(INDEX1,1)+1
      REMAIN(INDEX1,K+1)=SAMPLE(K)
      GO TO 300
      END IF
      IF (TYPE(K) .EQ. 13) THEN
      NOSAMP(ACTIVE(I),INT5)=NOSAMP(ACTIVE(I),INT5)+1
      ANGCOS=COS(3.1415926/180.*SAMPLE(K)/409.4*CALCON(I))
      ANGSIN=SIN(3.1415926/180.*SAMPLE(K)/409.4*CALCON(I))
      AVG(INDEX1,INT5)=AVG(INDEX1,INT5)+ANGSIN
      STDEV(INDEX1,INT5)=STDEV(INDEX1,INT5)+ANGSIN**2
      INDEX2=(INT5-1)/3+13
      AVG(INDEX1,INDEX2)=AVG(INDEX1,INDEX2)+ANGSIN
      STDEV(INDEX1,INDEX2)=STDEV(INDEX1,INDEX2)+ANGSIN**2
      AVG(INDEX1,17)=AVG(INDEX1,17)+ANGSIN
      STDEV(INDEX1,17)=STDEV(INDEX1,17)+ANGSIN**2
C
C WSPACE is a work space vector.
C
      WSPACE(2*INT5-1)=WSPACE(2*INT5-1)+ANGCOS
      WSPACE(2*INT5)=WSPACE(2*INT5)+ANGCOS**2
      WSPACE(2*((INT5-1)/3)+25)=WSPACE(2*((INT5-1)/3)+25)+ANGCOS
      WSPACE(2*((INT5-1)/3)+26)=WSPACE(2*((INT5-1)/3)+26)+ANGCOS**2
      WSPACE(33)=WSPACE(33)+ANGCOS
      WSPACE(34)=WSPACE(34)+ANGCOS**2
      END IF
 300  CONTINUE
      END IF
C
C Process data for all channels except radars, wind vanes, and UVW
C anemometers.
C
      IF (RTYPE(I).NE.13 .AND. ACTIVE(I).GE.11) THEN
      INDEX1=ACTIVE(I)-10
      DO 400 K=1,SAMPNO
      IF (IFIX((K-1)*SRATE(ACTIVE(I))/6.+STAMP).GT.TIME .AND. TYPE(K).EQ
     *.RTYPE(I)) THEN
      REMAIN(INDEX1,1)=REMAIN(INDEX1,1)+1
      REMAIN(INDEX1,K+1)=SAMPLE(K)
      GO TO 400
      END IF
      IF (TYPE(K) .EQ. RTYPE(I)) THEN
      NOSAMP(ACTIVE(I),INT5)=NOSAMP(ACTIVE(I),INT5)+1
      SUM=SAMPLE(K)/409.4*CALCON(I)+OFFSET(I)
      AVG(INDEX1,INT5)=AVG(INDEX1,INT5)+SUM
      STDEV(INDEX1,INT5)=STDEV(INDEX1,INT5)+SUM**2
      INDEX2=(INT5-1)/3+13
      AVG(INDEX1,INDEX2)=AVG(INDEX1,INDEX2)+SUM
      STDEV(INDEX1,INDEX2)=STDEV(INDEX1,INDEX2)+SUM**2
      AVG(INDEX1,17)=AVG(INDEX1,17)+SUM
      STDEV(INDEX1,17)=STDEV(INDEX1,17)+SUM**2
      END IF
 400  CONTINUE
      END IF
C
C
      CALL pass (ACTIVE(I),TIME,STATUS(ACTIVE(I)),HEAD,RADAR,REG)
      SAMPNO=6
      GO TO 185
      END IF
      END IF
C
C This section processes the UVW anemometers.
```

```
C
      IF (RTYPE(I) .EQ. 19) THEN
      DO 610 ANEM=1,NOUVW
      IF (ACTIVE(I) .EQ. UVW(ANEM,1)) GO TO 620
 610  CONTINUE
      GO TO 800
 620  CONTINUE
      CCON(1)=CALCON(I)
      DO 625 J=I,NOCHAN
      IF (ACTIVE(J) .EQ. UVW(ANEM,2)) CCON(2)=CALCON(J)
      IF (ACTIVE(J) .EQ. UVW(ANEM,3)) CCON(3)=CALCON(J)
 625  CONTINUE
C
C Process all UVW data for this anemometer for a 1 hr period.
C
      DO 630 K=1,17
      IF (K.LE.12) UVWFLG(K)=.FALSE.
      DO 635 J=1,3
      IF (K.LE.12) NOSAMP(UVW(ANEM,J),K)=0
      AVG(UVW(ANEM,J)-10,K)=0.
      STDEV(UVW(ANEM,J)-10,K)=0.
 635  CONTINUE
 630  CONTINUE
      DO 645 J=1,51
      VECTOR(ANEM,J)=0.
      STDVEC(ANEM,J)=0.
 645  CONTINUE
      DO 650 L=1,3
      TOTAL=0
      DO 660 K=1,12
C
C If any data remains from previous time period, process it first.
C
      IF (REMAIN(UVW(ANEM,L)-10,1).NE.0) THEN
      SAMPNO=REMAIN(UVW(ANEM,L)-10,1)
      DO 670 J=1,SAMPNO
      SAMPLE(J)=REMAIN(UVW(ANEM,L)-10,7-SAMPNO+J)
      TYPE(J)=RTYPE(I)
 670  CONTINUE
      REMAIN(UVW(ANEM,L)-10,1)=0
      STATUS(UVW(ANEM,L))=1
      STAMP=TIME+K*300-300
      IF (K.EQ.1) START(ANEM,L)=OLDSTR(ANEM,L)
      ELSE
      SAMPNO=6
      CALL pass (UVW(ANEM,L),TIME+K*300,STATUS(UVW(ANEM,L)),HEAD,RADAR,R
     *EG)
      IF ((K.EQ.1 .OR. START(ANEM,L).EQ.0) .AND. STATUS(UVW(ANEM,L)).EQ.
     *1) START(ANEM,L)=STAMP
      END IF
      IF (STATUS(UVW(ANEM,L)) .EQ. 1) UVWFLG(K)=.TRUE.
 680  IF (STATUS(UVW(ANEM,L)) .EQ. 1) THEN
      DO 690 J=1,SAMPNO
      IF (TYPE(J) .NE. RTYPE(I)) UVWFLG(K)=.FALSE.
      IF (IFIX((J-1)*SRATE(UVW(ANEM,L))/6.+STAMP).GT.TIME+K*300. .AND. TY
     *PE(J).EQ.RTYPE(I)) THEN
      REMAIN(UVW(ANEM,L)-10,1)=REMAIN(UVW(ANEM,L)-10,1)+1
      REMAIN(UVW(ANEM,L)-10,J+1)=SAMPLE(J)
      IF (K.EQ.12) OLDSTR(ANEM,L)=(6-REMAIN(UVW(ANEM,L)-10,1))*SRATE(UVW
     *(ANEM,L))/6.+STAMP
      GO TO 690
      END IF
      IF (TYPE(J) .EQ. RTYPE(I)) THEN
      NOSAMP(UVW(ANEM,L),K)=NOSAMP(UVW(ANEM,L),K)+1
      TOTAL=TOTAL+1
      UVWDAT(TOTAL,L)=SAMPLE(J)/409.4*CCON(L)
      END IF
```

```
  690  CONTINUE
       OLDSTP=STAMP
       CALL pass (UVW(ANEM,L),TIME+K*300,STATUS(UVW(ANEM,L)),HEAD,RADAR,R
      *EG)
       IF (STAMP .GT. OLDSTP) SRATE(UVW(ANEM,L))=STAMP-OLDSTP
       SAMPNO=6
       GO TO 680
       END IF
  660  CONTINUE
  650  CONTINUE
C
C If START is zero for all 3 components, no data was process for this period.
C Leave UVW anemometer section.
C
       IF (START(ANEM,1).EQ.0 .AND. START(ANEM,2).EQ.0 .AND. START(ANEM,3
      *).EQ.0) GO TO 800
C
C Compare data sets to determine how response factors are to be applied.
C If all data is not present, no response factors are applied and no vector
C sum is determined for THAT interval.
C
       TIME1=MAX0(START(ANEM,1),START(ANEM,2),START(ANEM,3))
       INCU=(TIME1-START(ANEM,1))/(SRATE(UVW(ANEM,1))/6)+1
       INCV=(TIME1-START(ANEM,2))/(SRATE(UVW(ANEM,2))/6)+1
       INCW=(TIME1-START(ANEM,3))/(SRATE(UVW(ANEM,3))/6)+1
       LL=0
       MM=0
       NN=0
       TOTAL=0
       DO 700 K=1,12
       NOZERO=0
       UVWCNT(K)=0
       LAST=MIN0(NOSAMP(UVW(ANEM,1),K),NOSAMP(UVW(ANEM,2),K),NOSAMP(UVW(A
      *NEM,3),K))
       IF (UVWFLG(K) .AND. LAST.GT.1) THEN
       DO 710 J=1,LAST
       LL=LL+INCU
       MM=MM+INCV
       NN=NN+INCW
       IF (UVWDAT(LL,1).EQ.0 .AND. UVWDAT(MM,2).EQ.0) THEN
       NOZERO=NOZERO+1
       GO TO 710
       END IF
       TOTAL=TOTAL+1
       IF (UVWDAT(LL,1).EQ.0. .OR. UVWDAT(MM,2).EQ.0.) THEN
       RATIO=0.
       ELSE
       RATIO=ABS(UVWDAT(LL,1)/UVWDAT(MM,2))
       IF (RATIO .GT. 1) RATIO=1./RATIO
       END IF
       INDEX=IFIX((RATIO*100+2)/5)+1
C
C Apply response factors.
C
       IF (ABS(UVWDAT(LL,1)).GE.ABS(UVWDAT(MM,2))) THEN
       UVWDAT(LL,1)=UVWDAT(LL,1)*LARGE(INDEX)
       UVWDAT(MM,2)=UVWDAT(MM,2)*SMALL(INDEX)
       ELSE IF (ABS(UVWDAT(LL,1)).LT.ABS(UVWDAT(MM,2))) THEN
       UVWDAT(LL,1)=UVWDAT(LL,1)*SMALL(INDEX)
       UVWDAT(MM,2)=UVWDAT(MM,2)*LARGE(INDEX)
       END IF
C
C Calculate the vector sum.
C
       SUM=SQRT(UVWDAT(LL,1)**2+UVWDAT(MM,2)**2+UVWDAT(NN,3)**2)
       INDEX=(K-1)*3+1
       VECTOR(ANEM,INDEX)=VECTOR(ANEM,INDEX)+SUM
```

```
      STDVEC(ANEM,INDEX)=STDVEC(ANEM,INDEX)+SUM**2
      INDEX=((K-1)/3)*3+37
      VECTOR(ANEM,INDEX)=VECTOR(ANEM,INDEX)+SUM
      STDVEC(ANEM,INDEX)=STDVEC(ANEM,INDEX)+SUM**2
      VECTOR(ANEM,49)=VECTOR(ANEM,49)+SUM
      STDVEC(ANEM,49)=STDVEC(ANEM,49)+SUM**2
C
C Calculate the azimuth angle and the elevation angle.
C
C Here's the azimuth angle...(For UVW anemometers that have positive signals
C for southerly, westerly, and rising winds)
C
      AZM=180./3.141592654*ATAN2(-UVWDAT(LL,1),-UVWDAT(MM,2))
      IF (AZM .LT. 0.) AZM=AZM+360.
      AZM=90.-AZM
      IF (AZM .LT. 0.) AZM=AZM+360.
C
C And here's the elevation angle using direction cosines...
C
      IF (SUM .EQ. 0.) THEN
      ELEV=0.
      ELSE
      ELEV=90.-ACOS(UVWDAT(NN,3)/SUM)*180./3.141592654
      END IF
C
C Save angle data for later averaging.
C
      INDEX=(K-1)*3+3
      ANGCOS=COS(AZM*3.141592654/180.)
      ANGSIN=SIN(AZM*3.141592654/180.)
      VECTOR(ANEM,INDEX)=VECTOR(ANEM,INDEX)+ELEV
      STDVEC(ANEM,INDEX)=STDVEC(ANEM,INDEX)+ELEV**2
      INDEX=((K-1)/3)*3+39
      VECTOR(ANEM,INDEX)=VECTOR(ANEM,INDEX)+ELEV
      STDVEC(ANEM,INDEX)=STDVEC(ANEM,INDEX)+ELEV**2
      VECTOR(ANEM,51)=VECTOR(ANEM,51)+ELEV
      STDVEC(ANEM,51)=STDVEC(ANEM,51)+ELEV**2
      INDEX=(K-1)*3+2
      VECTOR(ANEM,INDEX)=VECTOR(ANEM,INDEX)+ANGSIN
      STDVEC(ANEM,INDEX)=STDVEC(ANEM,INDEX)+ANGSIN**2
      INDEX=((K-1)/3)*3+38
      VECTOR(ANEM,INDEX)=VECTOR(ANEM,INDEX)+ANGSIN
      STDVEC(ANEM,INDEX)=STDVEC(ANEM,INDEX)+ANGSIN**2
      VECTOR(ANEM,50)=VECTOR(ANEM,50)+ANGSIN
      STDVEC(ANEM,50)=STDVEC(ANEM,50)+ANGSIN**2
C
C WSPACE is a work space vector.  (Holds cosine data for wind bearing.)
C
      WSPACE(2*K-1)=WSPACE(2*K-1)+ANGCOS
      WSPACE(2*K)=WSPACE(2*K)+ANGCOS**2
      WSPACE(2*((K-1)/3)+25)=WSPACE(2*((K-1)/3)+25)+ANGCOS
      WSPACE(2*((K-1)/3)+26)=WSPACE(2*((K-1)/3)+26)+ANGCOS**2
      WSPACE(33)=WSPACE(33)+ANGCOS
      WSPACE(34)=WSPACE(34)+ANGCOS**2
 710  CONTINUE
C
C Place the number of samples for the period in NOSAMP.
C
      UVWCNT(K)=J-1-NOZERO
C
C Calculate 5 and 15 min averages on vector sums.
C
      INDEX=(K-1)*3+1
      SUM=VECTOR(ANEM,INDEX)
      SUMSQ=STDVEC(ANEM,INDEX)
      VECTOR(ANEM,INDEX)=SUM/UVWCNT(K)
      STDVEC(ANEM,INDEX)=SQRT(ABS(SUMSQ-SUM**2/UVWCNT(K))/(UVWCNT(K)-1))
```

```
       INDEX=(K-1)*3+3
       SUM=VECTOR(ANEM,INDEX)
       SUMSQ=STDVEC(ANEM,INDEX)
       VECTOR(ANEM,INDEX)=SUM/UVWCNT(K)
       STDVEC(ANEM,INDEX)=SQRT(ABS(SUMSQ-SUM**2/UVWCNT(K))/(UVWCNT(K)-1))
       INDEX=(K-1)*3+2
       SUM=VECTOR(ANEM,INDEX)
       SUMSQ=STDVEC(ANEM,INDEX)
       SUM1=WSPACE(2*K-1)
       SUM1SQ=WSPACE(2*K)
       VECTOR(ANEM,INDEX)=180./3.141592654*ATAN2(SUM/UVWCNT(K),SUM1/UVWCN
      *T(K))
       IF (VECTOR(ANEM,INDEX).LT.0) VECTOR(ANEM,INDEX)=VECTOR(ANEM,INDEX)
      *+360.
       STDVEC(ANEM,INDEX)=SQRT(ABS(SUMSQ-SUM**2/UVWCNT(K))/(UVWCNT(K)-1)+
      *ABS(SUM1SQ-SUM1**2/UVWCNT(K))/(UVWCNT(K)-1))*57.14
C
C Calculate 15 min averages.
C
       IF (MOD(K,3) .EQ. 0) THEN
       IF (UVWFLG(K).AND.UVWFLG(K-1).AND.UVWFLG(K-2)) THEN
       INDEX2=((K-1)/3)*3+37
       SUM=VECTOR(ANEM,INDEX2)
       SUMSQ=STDVEC(ANEM,INDEX2)
       CNT15=UVWCNT(K)+UVWCNT(K-1)+UVWCNT(K-2)
       VECTOR(ANEM,INDEX2)=SUM/CNT15
       STDVEC(ANEM,INDEX2)=SQRT(ABS(SUMSQ-SUM**2/CNT15)/(CNT15-1))
       INDEX2=((K-1)/3)*3+39
       SUM=VECTOR(ANEM,INDEX2)
       SUMSQ=STDVEC(ANEM,INDEX2)
       VECTOR(ANEM,INDEX2)=SUM/CNT15
       STDVEC(ANEM,INDEX2)=SQRT(ABS(SUMSQ-SUM**2/CNT15)/(CNT15-1))
       INDEX2=((K-1)/3)*3+38
       SUM=VECTOR(ANEM,INDEX2)
       SUMSQ=STDVEC(ANEM,INDEX2)
       SUM1=WSPACE(2*((K-1)/3)+25)
       SUM1SQ=WSPACE(2*((K-1)/3)+26)
       VECTOR(ANEM,INDEX2)=180./3.141592654*ATAN2(SUM/CNT15,SUM1/CNT15)
       IF (VECTOR(ANEM,INDEX2).LT.0) VECTOR(ANEM,INDEX2)=VECTOR(ANEM,IN
      *DEX2)+360.
       STDVEC(ANEM,INDEX2)=SQRT(ABS(SUMSQ-SUM**2/CNT15)/(CNT15-1)+ABS(S
      *UM1SQ-SUM1**2/CNT15)/(CNT15-1))*57.14
       ELSE
       DO 770 M=1,3
       VECTOR(ANEM,((K-1)/3)*3+M+36)=10.**30
       STDVEC(ANEM,((K-1)/3)*3+M+36)=10.**30
 770   CONTINUE
       END IF
       END IF
C
C If data is not present replace with stars.
C
       ELSE
       DO 775 M=1,3
       VECTOR(ANEM,(K-1)*3+M)=10.**30
       STDVEC(ANEM,(K-1)*3+M)=10.**30
 775   CONTINUE
       END IF
 700   CONTINUE
C
C Calculate the hourly average.
C
       NUMBER=0
       DO 780 K=1,12
       IF (UVWFLG(K)) NUMBER=NUMBER+1
 780   CONTINUE
       IF (NUMBER .GE. 9) THEN
```

```
      SUM=VECTOR(ANEM,49)
      SUMSQ=STDVEC(ANEM,49)
      VECTOR(ANEM,49)=SUM/TOTAL
      STDVEC(ANEM,49)=SQRT(ABS(SUMSQ-SUM**2/TOTAL)/(TOTAL-1))
      SUM=VECTOR(ANEM,51)
      SUMSQ=STDVEC(ANEM,51)
      VECTOR(ANEM,51)=SUM/TOTAL
      STDVEC(ANEM,51)=SQRT(ABS(SUMSQ-SUM**2/TOTAL)/(TOTAL-1))
      SUM=VECTOR(ANEM,50)
      SUMSQ=STDVEC(ANEM,50)
      SUM1=WSPACE(33)
      SUM1SQ=WSPACE(34)
      VECTOR(ANEM,50)=180./3.141592654*ATAN2(SUM/TOTAL,SUM1/TOTAL)
      IF (VECTOR(ANEM,50).LT.0) VECTOR(ANEM,50)=VECTOR(ANEM,50)+360.
      STDVEC(ANEM,50)=SQRT(ABS(SUMSQ-SUM**2/TOTAL)/(TOTAL-1)+ABS(SUM1SQ-
     *SUM1**2/TOTAL)/(TOTAL-1))*57.14
      ELSE
      DO 790 M=1,3
      VECTOR(ANEM,48+M)=10.**30
      STDVEC(ANEM,48+M)=10.**30
 790  CONTINUE
      END IF
C
C Obtain individual component data for averages.
C
      DO 740 L=1,3
      DO 750 K=1,12
      IF (K.EQ.1) THEN
      LL=1
      LAST=NOSAMP(UVW(ANEM,L),1)
      ELSE
      LL=LAST+1
      LAST=LAST+NOSAMP(UVW(ANEM,L),K)
      END IF
      DO 760 J=LL,LAST
      INDEX1=UVW(ANEM,L)-10
      AVG(INDEX1,K)=AVG(INDEX1,K)+UVWDAT(J,L)
      STDEV(INDEX1,K)=STDEV(INDEX1,K)+UVWDAT(J,L)**2
      INDEX2=(K-1)/3+13
      AVG(INDEX1,INDEX2)=AVG(INDEX1,INDEX2)+UVWDAT(J,L)
      STDEV(INDEX1,INDEX2)=STDEV(INDEX1,INDEX2)+UVWDAT(J,L)**2
      AVG(INDEX1,17)=AVG(INDEX1,17)+UVWDAT(J,L)
      STDEV(INDEX1,17)=STDEV(INDEX1,17)+UVWDAT(J,L)**2
 760  CONTINUE
 750  CONTINUE
 740  CONTINUE
      END IF
C
C Determine if there is enough data to print averages.
C
 800  IF (NOSAMP(ACTIVE(I),INT5) .GT. 0) PRT=.TRUE.
C
C Determine if time to go to next day.
C
      NEXT=.TRUE.
      DO 180 J=1,64
      IF (STATUS(J) .NE. 0) NEXT=.FALSE.
      NEXT=.TRUE..AND.NEXT
 180  CONTINUE
      IF (NEXT) GO TO 190
      IF (RTYPE(I) .EQ. 19) GO TO 140
 200  CONTINUE
C
C Calculate Wind Vane averages.
C
C
C Here are the 5 min avgs.
```

```
C
 190  IF (RTYPE(I) .EQ. 13 .AND. PRT) THEN
      COUNT=0
      INDEX1=ACTIVE(I)-10
      DO 310 K=1,12
      COUNT=COUNT+NOSAMP(ACTIVE(I),K)
      IF (NOSAMP(ACTIVE(I),K) .GE. IFIX(300./(SRATE(ACTIVE(I))/6.)*3./4.
     *)) THEN
      SUM=AVG(INDEX1,K)
      SUMSQ=STDEV(INDEX1,K)
      SUM1=WSPACE(2*K-1)
      SUM1SQ=WSPACE(2*K)
      AVG(INDEX1,K)=180./3.1415926*ATAN2(SUM/NOSAMP(ACTIVE(I),K),SUM1/NO
     *SAMP(ACTIVE(I),K))
      IF (AVG(INDEX1,K) .LT. 0) AVG(INDEX1,K)=AVG(INDEX1,K)+360.
      STDEV(INDEX1,K)=SQRT(ABS(SUMSQ-SUM**2/NOSAMP(ACTIVE(I),K))/(NOSAMP
     *(ACTIVE(I),K)-1)+ABS(SUM1SQ-SUM1**2/NOSAMP(ACTIVE(I),K))/(NOSAMP(A
     *CTIVE(I),K)-1))*57.14
      ELSE
      AVG(INDEX1,K)=10.**30
      STDEV(INDEX1,K)=10.**30
      END IF
C
C Here are the 15 min averages.
C
      IF (MOD(K,3) .EQ. 0) THEN
      INDEX2=(K-1)/3+13
      IF (NOSAMP(ACTIVE(I),K)+NOSAMP(ACTIVE(I),K-1)+NOSAMP(ACTIVE(I),K-2
     *) .GE. IFIX(900./(SRATE(ACTIVE(I))/6.)*3./4.)) THEN
      INDEX2=(K-1)/3+13
      SUM=AVG(INDEX1,INDEX2)
      SUMSQ=STDEV(INDEX1,INDEX2)
      SUM1=WSPACE(2*((K-1)/3)+25)
      SUM1SQ=WSPACE(2*((K-1)/3)+26)
      TOTAL=NOSAMP(ACTIVE(I),K)+NOSAMP(ACTIVE(I),K-1)+NOSAMP(ACTIVE(I),K
     *-2)
      AVG(INDEX1,INDEX2)=180./3.1415926*ATAN2(SUM/TOTAL,SUM1/TOTAL)
      IF (AVG(INDEX1,INDEX2) .LT. 0) AVG(INDEX1,INDEX2)=AVG(INDEX1,INDEX
     *2)+360.
      STDEV(INDEX1,INDEX2)=SQRT(ABS(SUMSQ-SUM**2/TOTAL)/(TOTAL-1)+ABS(SU
     *M1SQ-SUM1**2/TOTAL)/(TOTAL-1))*57.14
      ELSE
      AVG(INDEX1,INDEX2)=10.**30
      STDEV(INDEX1,INDEX2)=10.**30
      END IF
      END IF
 310  CONTINUE
C
C And the hourly averages...
C
      IF (COUNT .GE. IFIX(3600./(SRATE(ACTIVE(I))/6.)*3./4.)) THEN
      SUM=AVG(INDEX1,17)
      SUMSQ=STDEV(INDEX1,17)
      SUM1=WSPACE(33)
      SUM1SQ=WSPACE(34)
      AVG(INDEX1,17)=180./3.1415926*ATAN2(SUM/COUNT,SUM1/COUNT)
      IF (AVG(INDEX1,17) .LT. 0) AVG(INDEX1,17)=AVG(INDEX1,17)+360.
      STDEV(INDEX1,17)=SQRT(ABS(SUMSQ-SUM**2/COUNT)/(COUNT-1)+ABS(SUM1SQ
     *-SUM1**2/COUNT)/(COUNT-1))*57.14
      ELSE
      AVG(INDEX1,17)=10.**30
      STDEV(INDEX1,17)=10.**30
      END IF
      END IF
 140  CONTINUE
C
C Print and calculate averages if required.
```

```
C
      IF (PRT) THEN
      CALL CALC (PERIOD,ACTIVE,NOCHAN,RADAVG,AVG,STDEV,SRATE,NOSAMP,CALI
     *B)
      CALL WRITER (PERIOD,ACTIVE,NOCHAN,RADAVG,AVG,STDEV,SRATE,NOSAMP,
     *BEGIN,DAYMAX,CALIB,HRMAX)
      WRITE (6,1)
      DO 155 L=1,64
      DO 195 K=1,12
      NOSAMP (L,K)=0
 195  CONTINUE
 155  CONTINUE
      END IF
      TIME=BEGIN+3600
      IF (.NOT.NEXT) GO TO 160
C
C Print daily maximums.
C
      WRITE (6,1)
      WRITE (6,220)
 220  FORMAT(' ',37X,'Daily Five Minute Average Maximums for each Active
     * Channel'/'+',37X,58('_'),//,38X,'Channel',20X,'Time',20X,'Value'/
     *'+',38X,7('_'),20X,4('_'),20X,5('_'),/)
      DO 240 J=1,NOCHAN
      HOUR=DAYMAX(ACTIVE(J),2)/3600
      MINUTE=(DAYMAX(ACTIVE(J),2)-HOUR*3600)/60
      SECOND=DAYMAX(ACTIVE(J),2)-HOUR*3600-MINUTE*60
      IF (ACTIVE(J) .LE. 10) THEN
      WRITE (6,250) ACTIVE(J),HOUR,MINUTE,SECOND,DAYMAX(ACTIVE(J),1)
 250  FORMAT(' ',41X,I2.2,20X,I2.2,':',I2.2,':',I2.2,13X,I4,1X,'vehicles
     *')
      ELSE
      WRITE (6,230) ACTIVE(J),HOUR,MINUTE,SECOND,DAYMAX(ACTIVE(J),1),
     *UNITS(J)
 230  FORMAT(' ',41X,I2.2,20X,I2.2,':',I2.2,':',I2.2,12X,F8.3,' ',A8)
      END IF
 240  CONTINUE
C
C Print hourly average maximums for the day.
C
      WRITE (6,1)
      WRITE (6,1220)
1220  FORMAT(' ',39X,'Daily Hourly Average Maximums for each Active Chan
     *nel'/'+',39X,53('_'),//,38X,'Channel',20X,'Time',20X,'Value'/
     *'+',38X,7('_'),20X,4('_'),20X,5('_'),/)
      DO 1240 J=1,NOCHAN
      HOUR=HRMAX(ACTIVE(J),2)/3600
      MINUTE=(HRMAX(ACTIVE(J),2)-HOUR*3600)/60
      SECOND=HRMAX(ACTIVE(J),2)-HOUR*3600-MINUTE*60
      IF (ACTIVE(J) .LE. 10) THEN
      WRITE (6,250) ACTIVE(J),HOUR,MINUTE,SECOND,HRMAX(ACTIVE(J),1)
      ELSE
      WRITE (6,230) ACTIVE(J),HOUR,MINUTE,SECOND,HRMAX(ACTIVE(J),1),
     *UNITS(J)
      END IF
1240  CONTINUE
      WRITE (6,1250)
1250  FORMAT(' ',///,15X,'NOTE:  All times in the maximums are ending ti
     *mes for the interval')
      WRITE (6,1)
C
C Close temporary files.  Prepare for next day's run.
C
      CALL split (0,EOF)
 100  CONTINUE
      CALL TIMER
      CALL FINIS
```

```
      CALL opener (2)
      STOP
      END
C
C
C This BLOCK DATA subprogram assigns all of the correction factors
C for UVW non-cosine response.
C
      BLOCK DATA
      COMMON /COR/ LARGE,SMALL
      REAL*4 LARGE(21),SMALL(21)
      DATA LARGE /1.,1.,1.,1.002,1.004,1.008,1.013,1.018,1.024,1.03,1.03
     *7,1.043,1.049,1.057,1.066,1.075,1.083,1.093,1.103,1.115,1.130/
      DATA SMALL /1.25,1.25,1.25,1.25,1.25,1.25,1.248,1.245,1.240,1.235,
     *1.228,1.22,1.212,1.203,1.193,1.183,1.173,1.163,1.152,1.141,1.130/
      END
C
C
C
C This subroutine is responsible for printing out the averages.
C
      SUBROUTINE WRITER (PERIOD,ACTIVE,NOCHAN,RADAVG,AVG,STDEV,SRATE,
     *NOSAMP,BEGIN,DAYMAX,CALIB,HRMAX)
      IMPLICIT INTEGER*4 (A-Z)
      INTEGER*4 ACTIVE(64),SRATE(64),NOSAMP(64,12),UVW(6,3),CAT(5)
      REAL*4 AVG(54,17),STDEV(54,17),RADAVG(10,5,34),SPEED,DAYMAX(64,2)
      REAL*4 TOTAL,VECTOR(6,51),STDVEC(6,51),SPD(5),PERCT(5),HRMAX(64,2)
      LOGICAL PRT,CALIB(64)
      INTEGER*2 RTYPE(64)
      CHARACTER*8 UNITS(64)
      CHARACTER*12 NAME(12)
      COMMON RTYPE,UNITS,NAME
      COMMON /RADINF/ MONTH,DATE,YEAR
      COMMON /UVWS/ VECTOR,STDVEC,NOUVW,UVW
      IF (INT5 .GT. 12) INT5=12
      TIME=BEGIN
      INT5=PERIOD
C
C Print the five minute averages.
C
      DO 10 I=1,INT5
      TIME=TIME+300
      HOUR=TIME/3600
      MINUTE=(TIME-HOUR*3600)/60
      SECOND=TIME-HOUR*3600-MINUTE*60
      WRITE (6,20) HOUR,MINUTE,SECOND
   20 FORMAT(' ',41X,'Average for five minute period ending at ',I2.2,':
     *',I2.2,':',I2.2,/)
C
C Print radar averages.
C
      PRT=.FALSE.
      DO 30 J=1,10
      IF (NOSAMP(J,I) .GT. 0) PRT=.TRUE.
   30 CONTINUE
      IF (PRT) THEN
      WRITE (6,40)
   40 FORMAT(' ',59X,'Traffic Data'/'+',59X,12('_'),/)
      SPEED=0.
      NOCARS=0
      DO 35 K=1,5
      CAT(K)=0
      SPD(K)=0.
   35 CONTINUE
      DO 50 J=1,10
      DO 60 K=1,5
      IF (IFIX(RADAVG(J,K,2*I-1)) .GT. 0 .AND. RADAVG(J,K,2*I-1).LT.10.*
```

```
     **20) THEN
      CAT(K)=CAT(K)+IFIX(RADAVG(J,K,2*I-1))
      SPD(K)=SPD(K)+RADAVG(J,K,2*I)*RADAVG(J,K,2*I-1)
      NOCARS=NOCARS+IFIX(RADAVG(J,K,2*I-1))
      SPEED=SPEED+RADAVG(J,K,2*I)*RADAVG(J,K,2*I-1)
      END IF
   60 CONTINUE
   50 CONTINUE
      IF (NOCARS .NE. 0) THEN
      SPEED=SPEED/NOCARS
      ELSE
      SPEED=0.
      END IF
      WRITE (6,70) NOCARS,SPEED
   70 FORMAT(' ',40X,'Total Traffic = ',I5,' vehicles averaging ',F4.1,'
     * mph',/,' ')
      WRITE (6,80)
   80 FORMAT(' ',55X,'Westbound Traffic'/'+',55X,17('_'),/,' ')
      WRITE (6,90)
   90 FORMAT(' ',25X,'Radar 1',15X,'Radar 3',15X,'Radar 5',15X,'Radar 7'
     *,15X,'Radar 9'/'+',25X,4(7('_'),15X),7('_'),/,24X,4('No.',5X,'Spd.
     *',10X),'No.',5X,'Spd'/'+',23X,4(3('_'),5X,4('_'),10X),3('_'),5X,4(
     *'_'))
      DO 100 K=1,5
      WRITE (6,110) K,(RADAVG(J,K,2*I-1),RADAVG(J,K,2*I),J=1,9,2)
  110 FORMAT(' ',9X,'Cat',I1,5(09X,I4,5X,F4.1))
  100 CONTINUE
      WRITE (6,120)
  120 FORMAT(' ',//,' ',55X,'Eastbound Traffic'/'+',55X,17('_'))
      WRITE (6,130)
  130 FORMAT(' ',25X,'Radar 2',15X,'Radar 4',15X,'Radar 6',15X,'Radar 8'
     *,15X,'Radar 10'/'+',25X,4(7('_'),15X),8('_'),/,24X,4('No.',5X,'Spd
     *.',10X),'No.',5X,'Spd'/'+',23X,4(3('_'),5X,4('_'),10X),3('_'),5X,4
     *('_'))
      DO 140 K=1,5
      WRITE (6,110) K,(RADAVG(J,K,2*I-1),RADAVG(J,K,2*I),J=2,10,2)
  140 CONTINUE
      WRITE (6,150)
  150 FORMAT(' ',//,' ')
C
C This section provides the catagory breakdowns.
C
      WRITE (6,245)
  245 FORMAT(' ',51X,'Vehicle Classification Summary'/'+',51X,30('_'),//
     *,' ',17X,'CAT 1',20X,'CAT 2',20X,'CAT 3',20X,'CAT 4',20X,'CAT 5'/'
     *+',17X,4(5('_'),20X),5('_'),//,' ',10X,4('Spd.',03X,'Total',05X,'%
     *',07X),'Spd.',03X,'Total',05X,'%'/'+',10X,4(4('_'),03X,5('_'),05X,
     *'_',07X),4('_'),03X,5('_'),05X,'_',/,' ')
      DO 265 K=1,5
      IF (CAT(K).NE.0) THEN
      SPD(K)=SPD(K)/CAT(K)
      ELSE
      SPD(K)=0.
      END IF
      IF (NOCARS.NE.0) THEN
      PERCT(K)=FLOAT(CAT(K))/FLOAT(NOCARS)*100.
      ELSE
      PERCT(K)=10.**30
      END IF
  265 CONTINUE
      WRITE (6,255) (SPD(K),CAT(K),PERCT(K),K=1,5)
  255 FORMAT(' ',10X,4(F4.1,03X,I4,04X,F4.1,06X),F4.1,03X,I4,04X,F4.1,//
     */,' ')
      END IF
C
C Determine MAX values for radars.
C
```

```
      DO 180 J=1,10
      DO 190 L=1,12
      TOTAL=0.
      DO 200 K=1,5
      TOTAL=TOTAL+RADAVG(J,K,2*L-1)
  200 CONTINUE
      IF ((DAYMAX(J,1).LT.TOTAL) .AND. TOTAL.LT.10.**20) THEN
      DAYMAX(J,1)=TOTAL
      DAYMAX(J,2)=BEGIN+L*300
      END IF
  190 CONTINUE
  180 CONTINUE
C
C
C Print 5 min avgs for regular channels.
C
      PRT=.FALSE.
      DO 210 J=11,64
      IF (NOSAMP(J,I) .GT. 0) PRT=.TRUE.
  210 CONTINUE
      IF (PRT) THEN
C
C Print the averages for the regular channels.
C
      WRITE (6,600)
  600 FORMAT(' ',57X,'Regular Channels'/'+',57X,16('_'),///,' ')
      WRITE (6,610)
  610 FORMAT(' ','Channel',06X,'Instrument',13X,'Value',13X,'Units',12X
     *,'Channel',06X,'Instrument',13X,'Value',13X,'Units'/'+',7('_'),06
     *X,10('_'),13X,5('_'),13X,5('_'),12X,7('_'),06X,10('_'),13X,5('_')
     *,13X,5('_'),/,' ')
      K=(NOCHAN-10)/2
      DO 620 J=11,K+10
      WRITE (6,630) ACTIVE(J),NAME(J),AVG(ACTIVE(J)-10,I),STDEV(ACTIVE(J
     *)-10,I),UNITS(J),ACTIVE(J+K),NAME(J+K),AVG(ACTIVE(J+K)-10,I),STDEV
     *(ACTIVE(J+K)-10,I),UNITS(J+K)
  630 FORMAT(' ',02X,I2.2,08X,A12,5X,F8.3,'+/-',F8.3,05X,A8,12X,I2.2,08X
     *,A12,5X,F8.3,'+/-',F8.3,05X,A8)
  620 CONTINUE
      IF ((J-1+K) .LT. NOCHAN) WRITE (6,630) ACTIVE(NOCHAN),NAME(NOCHAN)
     *,AVG(ACTIVE(NOCHAN)-10,I),STDEV(ACTIVE(NOCHAN)-10,I),UNITS(NOCHAN)
      WRITE (6,640)
  640 FORMAT(' ',//,' ')
C
C Print the vector sums for the UVW anemometers.
C
      WRITE (6,650)
  650 FORMAT(' ',54X,'UVW Vector Calculations'/'+',54X,23('_'),//)
      WRITE (6,660)
  660 FORMAT(' ',24X,'UVW',06X,'Channels',06X,'Vector Sum (mph)',05X,'Be
     *aring Deg AZ',05X,'Elevation Angle (Deg)'/'+',24X,3('_'),06X,8('_'
     *),06X,16('_'),05X,14('_'),05X,21('_'),/,' ')
      DO 670 ANEM=1,NOUVW
      WRITE (6,680) ANEM,(UVW(ANEM,L),L=1,3),(VECTOR(ANEM,(I-1)*3+M),STD
     *VEC(ANEM,(I-1)*3+M),M=1,3)
  680 FORMAT(' ',25X,I2.2,04X,3(I3.2,1X),05X,F5.2,'+/-',F5.2,07X,F5.1,'+
     */-',F5.1,10X,F5.1,'+/-',F5.1)
  670 CONTINUE
      WRITE (6,645)
  645 FORMAT(' ',//,' ',132('*'),////,' ')
      END IF
C
   10 CONTINUE
C
C Determine MAX values for regular channels
C
      DO 300 J=11,NOCHAN
```

```
      DO 310 L=1,12
      IF (DAYMAX(ACTIVE(J),1).LT.AVG(ACTIVE(J)-10,L) .AND. AVG(ACTIVE(J)
     *-10,L).LT.10.**20) THEN
      DAYMAX(ACTIVE(J),1)=AVG(ACTIVE(J)-10,L)
      DAYMAX(ACTIVE(J),2)=BEGIN+L*300
      END IF
  310 CONTINUE
  300 CONTINUE
C
C
C Now print 15 min averages.
C
 8000 TIME=BEGIN
      DO 1000 I=1,INT5/3
      TIME=TIME+900
      HOUR=TIME/3600
      MINUTE=(TIME-HOUR*3600)/60
      SECOND=TIME-HOUR*3600-MINUTE*60
      WRITE (6,1020) HOUR,MINUTE,SECOND
 1020 FORMAT(' ',42X,'Average for 15 minute period ending at ',I2.2,':',
     *I2.2,':',I2.2,/)
C
C Print radar data.
C
      PRT=.FALSE.
      DO 1030 J=1,10
      IF (NOSAMP(J,3*I-2)+NOSAMP(J,3*I-1)+NOSAMP(J,3*I) .GT. 0) PRT=.TRU
     *E.
 1030 CONTINUE
      IF (PRT) THEN
      WRITE (6,40)
      SPEED=0.
      NOCARS=0
      DO 1035 K=1,5
      CAT(K)=0
      SPD(K)=0.
 1035 CONTINUE
      DO 1040 J=1,10
      DO 1050 K=1,5
      IF (IFIX(RADAVG(J,K,2*(I-1)+25)).GT.0 .AND. RADAVG(J,K,2*(I-1)+26)
     *.LT.10.**20) THEN
      CAT(K)=CAT(K)+IFIX(RADAVG(J,K,2*(I-1)+25))
      SPD(K)=SPD(K)+RADAVG(J,K,2*(I-1)+25)*RADAVG(J,K,2*(I-1)+26)
      NOCARS=NOCARS+IFIX(RADAVG(J,K,2*(I-1)+25))
      SPEED=SPEED+RADAVG(J,K,2*(I-1)+25)*RADAVG(J,K,2*(I-1)+26)
      END IF
 1050 CONTINUE
 1040 CONTINUE
      IF (NOCARS .NE. 0) THEN
      SPEED=SPEED/NOCARS
      ELSE
      SPEED=0.
      END IF
      WRITE (6,70) NOCARS,SPEED
      WRITE (6,80)
      WRITE (6,90)
      DO 1060 K=1,5
      WRITE (6,110) K,(RADAVG(J,K,2*(I-1)+25),RADAVG(J,K,2*(I-1)+26),J=1
     *,9,2)
 1060 CONTINUE
      WRITE (6,120)
      WRITE (6,130)
      DO 1070 K=1,5
      WRITE (6,110) K,(RADAVG(J,K,2*(I-1)+25),RADAVG(J,K,2*(I-1)+26),J=2
     *,10,2)
 1070 CONTINUE
      WRITE (6,150)
```

```
C
C This section provides the catagory breakdowns.
C
      WRITE (6,245)
      DO 1265 K=1,5
      IF (CAT(K).NE.0) THEN
      SPD(K)=SPD(K)/CAT(K)
      ELSE
      SPD(K)=0.
      END IF
      IF (NOCARS.NE.0) THEN
      PERCT(K)=FLOAT(CAT(K))/FLOAT(NOCARS)*100.
      ELSE
      PERCT(K)=10.**30
      END IF
1265  CONTINUE
      WRITE (6,255) (SPD(K),CAT(K),PERCT(K),K=1,5)
      END IF
C
C Now 15 min averages for other channels.
C
      PRT=.FALSE.
      DO 1090 J=11,64
      IF (NOSAMP(J,3*I-2)+NOSAMP(J,3*I-1)+NOSAMP(J,3*I) .GT. 0) PRT=.TRU
     *E.
1090  CONTINUE
      IF (PRT) THEN
C
C Print the averages for the regular channels.
C
      WRITE (6,600)
      WRITE (6,610)
      K=(NOCHAN-10)/2
      DO 1100 J=11,K+10
      WRITE (6,630) ACTIVE(J),NAME(J),AVG(ACTIVE(J)-10,I+12),STDEV(ACTIV
     *E(J)-10,I+12),UNITS(J),ACTIVE(J+K),NAME(J+K),AVG(ACTIVE(J+K)-10,I+
     *12),STDEV(ACTIVE(J+K)-10,I+12),UNITS(J+K)
1100  CONTINUE
      IF ((J-1+K) .LT. NOCHAN) WRITE (6,630) ACTIVE(NOCHAN),NAME(NOCHAN)
     *,AVG(ACTIVE(NOCHAN)-10,I+12),STDEV(ACTIVE(NOCHAN)-10,I+12),UNITS(N
     *OCHAN)
      WRITE (6,640)
C
C Print the vector sums for the UVW anemometers.
C
      WRITE (6,650)
      WRITE (6,660)
      DO 1110 ANEM=1,NOUVW
      WRITE (6,680) ANEM,(UVW(ANEM,L),L=1,3),(VECTOR(ANEM,((I*3-1)/3)*3+
     *M+36),STDVEC(ANEM,((I*3-1)/3)*3+M+36),M=1,3)
1110  CONTINUE
      WRITE (6,645)
      END IF
C
 1000  CONTINUE
C
C Print the hourly averages.
C
9000  TIME=BEGIN
      TIME=TIME+3600
C
C Print the radar averages.
C
      HOUR=TIME/3600
      MINUTE=(TIME-HOUR*3600)/60
      SECOND=TIME-HOUR*3600-MINUTE*60
      WRITE (6,2000) HOUR,MINUTE,SECOND
```

```
2000  FORMAT(' ',44X,'Average for hourly period ending at ',I2.2,':',I2.
     *2,':',I2.2,/)
      PRT=.FALSE.
      DO 2010 J=1,10
      DO 2020 I=1,INT5
      IF (NOSAMP(J,I) .GT. 0) PRT=.TRUE.
2020  CONTINUE
2010  CONTINUE
      IF (PRT) THEN
      WRITE (6,40)
      SPEED=0.
      NOCARS=0
      DO 2035 K=1,5
      CAT(K)=0
      SPD(K)=0.
2035  CONTINUE
      DO 2030 J=1,10
      DO 2040 K=1,5
      IF (IFIX(RADAVG(J,K,33)).GT.0 .AND. RADAVG(J,K,34).LT.10.**20) THE
     *N
      CAT(K)=CAT(K)+IFIX(RADAVG(J,K,33))
      SPD(K)=SPD(K)+RADAVG(J,K,33)*RADAVG(J,K,34)
      NOCARS=NOCARS+IFIX(RADAVG(J,K,33))
      SPEED=SPEED+RADAVG(J,K,33)*RADAVG(J,K,34)
      END IF
2040  CONTINUE
2030  CONTINUE
      IF (NOCARS .NE. 0) THEN
      SPEED=SPEED/NOCARS
      ELSE
      SPEED=0.
      END IF
      WRITE (6,70) NOCARS,SPEED
      WRITE (6,80)
      WRITE (6,90)
      DO 2050 K=1,5
      WRITE (6,110) K,(RADAVG(J,K,33),RADAVG(J,K,34),J=1,9,2)
2050  CONTINUE
      WRITE (6,120)
      WRITE (6,130)
      DO 2060 K=1,5
      WRITE (6,110) K,(RADAVG(J,K,33),RADAVG(J,K,34),J=2,10,2)
2060  CONTINUE
      WRITE (6,150)
C
C This section provides the catagory breakdowns.
C
      WRITE (6,245)
      DO 2265 K=1,5
      IF (CAT(K).NE.0) THEN
      SPD(K)=SPD(K)/CAT(K)
      ELSE
      SPD(K)=0.
      END IF
      IF (NOCARS.NE.0) THEN
      PERCT(K)=FLOAT(CAT(K))/FLOAT(NOCARS)*100.
      ELSE
      PERCT(K)=10.**30
      END IF
2265  CONTINUE
      WRITE (6,255) (SPD(K),CAT(K),PERCT(K),K=1,5)
      END IF
C
C Determine hourly maximums for radars.
C
      DO 2300 J=1,10
      TOTAL=0.
```

```
      DO 2310 K=1,5
      TOTAL=TOTAL+RADAVG(J,K,33)
2310  CONTINUE
      IF ((HRMAX(J,1).LT.TOTAL) .AND. TOTAL.LT.10.**20) THEN
      HRMAX(J,1)=TOTAL
      HRMAX(J,2)=BEGIN+3600
      END IF
2300  CONTINUE
C
C Now regular channels
C
      PRT=.FALSE.
      DO 2090 J=11,64
      DO 2080 I=1,INT5
      IF (NOSAMP(J,I) .GT. 0) PRT=.TRUE.
2080  CONTINUE
2090  CONTINUE
      IF (PRT) THEN
C
C Print the averages for the regular channels.
C
      WRITE (6,600)
      WRITE (6,610)
      K=(NOCHAN-10)/2
      DO 2100 J=11,K+10
      WRITE (6,630) ACTIVE(J),NAME(J),AVG(ACTIVE(J)-10,17),STDEV(ACTIVE(
     *J)-10,17),UNITS(J),ACTIVE(J+K),NAME(J+K),AVG(ACTIVE(J+K)-10,17),ST
     *DEV(ACTIVE(J+K)-10,17),UNITS(J+K)
2100  CONTINUE
      IF ((J-1+K) .LT. NOCHAN) WRITE (6,630) ACTIVE(NOCHAN),NAME(NOCHAN)
     *,AVG(ACTIVE(NOCHAN)-10,17),STDEV(ACTIVE(NOCHAN)-10,17),UNITS(NOCHA
     *N)
      WRITE (6,640)
C
C Print the vector sums for the UVW anemometers.
C
      WRITE (6,650)
      WRITE (6,660)
      DO 2110 ANEM=1,NOUVW
      WRITE (6,680) ANEM,(UVW(ANEM,L),L=1,3),(VECTOR(ANEM,48+M),STDVEC(A
     *NEM,48+M),M=1,3)
2110  CONTINUE
      WRITE (6,645)
      END IF
C
C Determine hourly maximums for regular channels.
C
      DO 2320 J=11,NOCHAN
      IF (HRMAX(ACTIVE(J),1).LT.AVG(ACTIVE(J)-10,17).AND.AVG(ACTIVE(J)-1
     *0,17).LT.10.**20) THEN
      HRMAX(ACTIVE(J),1)=AVG(ACTIVE(J)-10,17)
      HRMAX(ACTIVE(J),2)=BEGIN+3600
      END IF
2320  CONTINUE
      RETURN
      END
C
C
C
C This subroutine calculates the averages for several of the channels.
C
      SUBROUTINE CALC (INT5,ACTIVE,NOCHAN,RADAVG,AVG,STDEV,SRATE,
     *NOSAMP,CALIB)
      IMPLICIT INTEGER*4 (A-Z)
      REAL*4 RADAVG(10,5,34),AVG(54,17),STDEV(54,17),SUM,SUMSQ
      REAL*8 SPEED
      INTEGER*4 ACTIVE(64),SRATE(64),NOSAMP(64,12)
```

```
      CHARACTER*8 UNITS(64)
      INTEGER*2 RTYPE(64)
      LOGICAL CALIB(64)
      COMMON RTYPE,UNITS
C
C Calculate radar averages.
C
      DO 10 J=1,10
      COUNT=0
      DO 20 K=1,5
      DO 30 L=1,12
      IF (K .EQ. 1) COUNT=COUNT+NOSAMP(J,L)
C
C Calculate 5 min avgs.
C
      IF (NOSAMP(J,L).GE.3 .AND. RADAVG(J,K,L*2-1).GT.0) THEN
      RADAVG(J,K,L*2)=RADAVG(J,K,L*2)/RADAVG(J,K,L*2-1)/409.4*20.*SQRT(2
     *.)
      ELSE IF (IFIX(RADAVG(J,K,L*2-1)).EQ.0 .AND. NOSAMP(J,L).GE.3) THEN
      RADAVG(J,K,L*2)=0.
      ELSE
      RADAVG(J,K,L*2)=10.**30
      RADAVG(J,K,L*2-1)=10.**30
      END IF
C
C Calculate 15 min average.
C
      IF (MOD(L,3) .EQ. 0) THEN
      IF (NOSAMP(J,L)+NOSAMP(J,L-1)+NOSAMP(J,L-2) .GE. 11) THEN
      IF (IFIX(RADAVG(J,K,2*(L/3-1)+25)) .NE. 0) THEN
      RADAVG(J,K,2*(L/3-1)+26)=RADAVG(J,K,2*(L/3-1)+26)/RADAVG(J,K,2*(L/
     *3-1)+25)/409.4*20.*SQRT(2.)
      ELSE
      RADAVG(J,K,2*(L/3-1)+26)=0.
      END IF
      ELSE
      RADAVG(J,K,2*(L/3-1)+25)=10.**30
      RADAVG(J,K,2*(L/3-1)+26)=10.**30
      END IF
      END IF
  30  CONTINUE
C
C Calculate hourly averages.
C
      IF (COUNT .GE. 45) THEN
      IF (IFIX(RADAVG(J,K,33)) .NE. 0) THEN
      RADAVG(J,K,34)=RADAVG(J,K,34)/RADAVG(J,K,33)/409.4*20.*SQRT(2.)
      ELSE
      RADAVG(J,K,34)=0.
      END IF
      ELSE
      RADAVG(J,K,33)=10.**30
      RADAVG(J,K,34)=10.**30
      END IF
  20  CONTINUE
  10  CONTINUE
C
C Calculate averages for regular channels.
C
      DO 40 I=11,NOCHAN
      IF (SRATE(ACTIVE(I)) .EQ. 0) GO TO 40
C
C Average data for all channels except radars and wind vanes.
C
      IF (RTYPE(I).NE.13 .AND. ACTIVE(I).GE.11) THEN
      COUNT=0
      DO 50 K=1,12
```

```
      COUNT=COUNT+NOSAMP(ACTIVE(I),K)
      IF (NOSAMP(ACTIVE(I),K) .GE. IFIX(300./(SRATE(ACTIVE(I))/6.)*3./4.
     *)) THEN
      SUM=AVG(ACTIVE(I)-10,K)
      SUMSQ=STDEV(ACTIVE(I)-10,K)
      AVG(ACTIVE(I)-10,K)=SUM/NOSAMP(ACTIVE(I),K)
      STDEV(ACTIVE(I)-10,K)=SQRT(ABS(SUMSQ-SUM**2/NOSAMP(ACTIVE(I),K))/(
     *NOSAMP(ACTIVE(I),K)-1))
      ELSE
      AVG(ACTIVE(I)-10,K)=10.**30
      STDEV(ACTIVE(I)-10,K)=10.**30
      END IF
C
C Here are the 15 min averages.
C
      IF (MOD(K,3) .EQ. 0) THEN
      IF (NOSAMP(ACTIVE(I),K)+NOSAMP(ACTIVE(I),K-1)+NOSAMP(ACTIVE(I),K-2
     *) .GE. IFIX(900./(SRATE(ACTIVE(I))/6.)*3./4.)) THEN
      SUM=AVG(ACTIVE(I)-10,(K-1)/3+13)
      SUMSQ=STDEV(ACTIVE(I)-10,(K-1)/3+13)
      TOTAL=NOSAMP(ACTIVE(I),K)+NOSAMP(ACTIVE(I),K-1)+NOSAMP(ACTIVE(I),K
     *-2)
      AVG(ACTIVE(I)-10,(K-1)/3+13)=SUM/TOTAL
      STDEV(ACTIVE(I)-10,(K-1)/3+13)=SQRT(ABS(SUMSQ-SUM**2/TOTAL)/(TOTAL
     *-1))
      ELSE
      AVG(ACTIVE(I)-10,(K-1)/3+13)=10.**30
      STDEV(ACTIVE(I)-10,(K-1)/3+13)=10.**30
      END IF
      END IF
  50  CONTINUE
C
C Here are the hourly averages...
C
      IF (COUNT .GE. IFIX(3600./(SRATE(ACTIVE(I))/6.)*3./4.)) THEN
      SUM=AVG(ACTIVE(I)-10,17)
      SUMSQ=STDEV(ACTIVE(I)-10,17)
      AVG(ACTIVE(I)-10,17)=SUM/COUNT
      STDEV(ACTIVE(I)-10,17)=SQRT(ABS(SUMSQ-SUM**2/COUNT)/(COUNT-1))
      ELSE
      AVG(ACTIVE(I)-10,17)=10.**30
      STDEV(ACTIVE(I)-10,17)=10.**30
      END IF
      END IF
  40  CONTINUE
      RETURN
      END
```

```
/* These are all of the C programs used in conjunction with the FORTRAN
   SETC program. */


#include <stdio.h>

FILE *tmpfile (), *tmp_point[65], *fopen (), *input;

#define         NO_CHAN         64              /* Total number of possible
                                                   channels on A/D */

unsigned char record[220];

/* This structure defines the beginning for each record.  It is the same
   on all records. */
struct general_info
 {
   long int time;
   short int channel;
   short int year;
   short int day;
   char name[10];
 };

/* This structure defines the data in the radar records. */
struct rad_params
 {
   short int type;
   short int no_vehicles[5];
   long int speed_sum[5];
 };

/* This structure defines the data in the regular channel records. */
struct reg_params
 {
   short int type[6];
   short int sample[6];
 };

/* This structure defines the data in the console messages. */
struct
 {
   short type;
   unsigned char ASCII[82];
 } console;

/* This routine will skip to a specified day if desired. */
skipper (day, offset)
  int *day;
  long *offset;
 {
   unsigned char data[220];
   int read_day, length;
   fseek (input, *offset, 0);
   fgets (data, 220, input);
   while ( fgets (data, 220, input) != NULL )
    {
      sscanf (data + 20, "%4x", &read_day);
      if ( read_day == *day )
        {
          sscanf (data, "%4x", &length);
          length *= 2;
          ++length;
          if ( fseek (input, -length, 1) != 0 )
            {
              fprintf (stderr, "Could not fseek in skipper.\n");
              exit (1);
```

```
            }
          return;
        }
    }
   fprintf (stderr, "Illegal offset for skipper.  End of file encountered.\n");
   exit (1);
 }


/* This routine opens the raw data file. */
opener (option)
 int *option;
 {
    if (*option == 1)
     {
      if ( (input = fopen ("/users/hlavinka/balcones/setaout/setb.out",
                           "r")) == NULL )
        {
          fprintf (stderr, "Input data could not be opened successfully.\n");
          exit (1);
        }
     }
    else if (*option == 2)
      fclose (input);
 }


/* This routine splits the large data file into small temporary files for
   each channel for each day.  After all averages have been calculated for
   the day, the files are closed and thus deleted. */

split (request, end_of_file)
  int *request;   /* This determines if files are split or closed. */
  int *end_of_file;  /* Determines when at EOF in raw data file. */
 {
   short int channel;
   long int date, old_date, length;
   int n = 0;
   char *get_record;

/* If request = 1 then split file. */
   if ( *request )
     {
      *end_of_file = 0;

/* Null all temporary file pointers. */
      for ( n = 0; n <= NO_CHAN; ++n )
        tmp_point[n] = NULL;
      fgets (record, 220, input);
      sscanf (record + 4, "%4hx", &channel);
      sscanf (record + 16, "%8lx", &old_date);
      if ( (tmp_point[channel] = tmpfile () ) == NULL )
        {
          fprintf (stderr, "Console message temp file not opened.\n");
          exit (1);
        }
      while ( (get_record = fgets (record, 220, input) ) != NULL )
        {
          sscanf (record, "%4x", &length);
          sscanf (record + 4, "%4hx", &channel);
          sscanf (record + 16, "%8lx", &date);
          if ( tmp_point[channel] == NULL )
            {
              if ( (tmp_point[channel] = tmpfile () ) == NULL &&
                        channel <= NO_CHAN )
                {
                  fprintf (stderr, "Temp file not opened.  Channel = %d\n",
```

```
                                channel);
                        exit (1);
                      }
                  }
              if ( date == old_date && channel <= NO_CHAN )
                fputs (record, tmp_point[channel]);
              else if ( date != old_date )
                break;
            }
          length *= 2;
          ++length;
          if ( fseek (input, -length, 1) != 0 )
            {
              fprintf (stderr, "fseek could not backup a record.\n");
              exit (1);
            }
          for ( n = 0; n <= NO_CHAN; ++n )
            {
              if ( tmp_point[n] != NULL )
                rewind (tmp_point[n]);
            }
          if ( get_record == NULL )
            *end_of_file = 1;

        }
      else
        {
          for ( n = 0; n <= NO_CHAN; ++n )
            {
              if ( tmp_point[n] != NULL )
                {
                  if ( fclose (tmp_point[n]) != 0 )
                    {
                      fprintf (stderr, "Could not close a file properly n = %d.\n",
                                        n );
                      exit (1);
                    }
                }
            }
        }
    }


/* This routine determines the sample rates of each channel.  The returned
   information is the time between subsequent records for all channels. */
sample_rate (srate)
 int *srate;
 {
   int n, time1, time2, rec_size;
   for ( n = 1; n <= NO_CHAN; ++n )
     {
       if ( n <= 10 )
         rec_size = -105;
       else if ( n >= 11 )
         rec_size = -89;
       if ( tmp_point[n] != NULL )
         {
           if (fgets (record, 220, tmp_point[n]) != NULL)
             {
               sscanf (record + 8, "%8lx", &time1);
               if (fgets (record, 220, tmp_point[n]) != NULL)
                 {
                   sscanf (record + 8, "%8lx", &time2);
                   *(srate + n - 1) = time2 - time1;
                   if ( fseek (tmp_point[n], 2 * rec_size, 1) != 0 )
                     {
                       fprintf (stderr, "Could not fseek in srate.\n");
```

```
                    exit (1);
                  }
              }
            else
              {
                if ( fseek(tmp_point[n], rec_size, 1) != 0 )
                  {
                    fprintf (stderr, "Could not fseek in srate.\n");
                    exit (1);
                  }
              }
          }
      }
    else
      *(srate + n - 1) = 0;
  }
}


/* This routine prints the date for each run.  It is called directly
   from FORTRAN before any data for the day has been processed. */
print_date (year, date, month, day_of_month, two_digit_year)
 int *year, *date, *month, *day_of_month, *two_digit_year;
 {
   char *convert ();
   struct general_info header;
   fgets (record, 220, tmp_point[0]);

/* Get date out of first console record. */
   sscanf (record + 16, "%4hx%4hx", &header.year, &header.day);
   fprintf (stdout, " Data for %s\n\n", convert (header.year, header.day, month,
     day_of_month, two_digit_year) );
   rewind (tmp_point[0]);
   *year = header.year;
   *date = header.day;
 }


/* This routine lists ASCII console messages to standard out. */
list (time_stamp)
 long int time_stamp;
 {
   struct
    {
      short int hour;
      short int minute;
      short int second;
    } time;
      time.hour = time_stamp / 3600;
      time.minute = (time_stamp - time.hour * 3600) / 60;
      time.second = time_stamp - time.hour * 3600 - time.minute * 60;
      fprintf (stdout, " %02d:%02d:%02d -- %s\n", time.hour, time.minute,
           time.second, console.ASCII);
      fflush (stdout);
 }


/* This function converts the Julian dates to calendar dates.  It returns
   a pointer to a NULL ended string containing the date. */
   char calendar_date[14];

   char *convert (year, day, month, day_of_month, two_digit_year)
   unsigned short int year, day;
   int *month, *day_of_month, *two_digit_year;
    {
      static struct
```

```
      {
        unsigned short int number_of_days;
        char name[4];
      } months[12] = { { 31, {'J', 'a', 'n'} }, { 28, {'F', 'e', 'b'} },
                       { 31, {'M', 'a', 'r'} }, { 30, {'A', 'p', 'r'} },
                       { 31, {'M', 'a', 'y'} }, { 30, {'J', 'u', 'n'} },
                       { 31, {'J', 'u', 'l'} }, { 31, {'A', 'u', 'g'} },
                       { 30, {'S', 'e', 'p'} }, { 31, {'O', 'c', 't'} },
                       { 30, {'N', 'o', 'v'} }, { 31, {'D', 'e', 'c'} } };
      int i, sum = 0, date;

/* Check if the given year is a leap year. */
      if ( (year % 4 == 0 && year % 100 != 0) || year % 400 == 0 )
        months[1].number_of_days = 29;    /* Is a leap year. */
      else
        months[1].number_of_days = 28;    /* Not a leap year. */

/* Determine date. */
      for ( i = 0; i < 12; ++i )
        {
          sum += months[i].number_of_days;
          if ( sum >= day )
            break;
        }
      date = day - (sum - months[i].number_of_days);
      sprintf (calendar_date, "%s. %02d, %04d", months[i].name, date, year);
      *month = i + 1;
      *day_of_month = date;
      *two_digit_year = year - 1900;
      return (calendar_date);
    }


/* This routine passes data back to the FORTRAN program.  The FORTRAN program
   then does all the calculations. */
pass (chan, time_end, status, header, radar, regular)
 long int *chan, *status;
 long int *time_end;
 struct general_info *header;
 struct reg_params *regular;
 struct rad_params *radar;
 {
   long int rec_size, time_stamp;
   short unsigned int character;
   int n;
   unsigned char BS = 0x08;
   unsigned char *ascii_pointer = console.ASCII;

/* Set up rec_size for hex ascii records. */
   if ( *chan >= 11 )
     rec_size = 89;      /* Regular records */
   else if ( *chan <= 10 && *chan > 0 )
     rec_size = 105;     /* Radar records */
   else if ( *chan == 0 )
     rec_size = 209;     /* Console messages */

/* Read records from time begin to time end. */
   if ( tmp_point[*chan] == NULL )
     {
       *status = 0;
       return;
     }
   if ( fgets (record, 220, tmp_point[*chan]) == NULL )
     *status = 0;
   else
     {
```

```
        *status = 1;
        sscanf (record + 8, "%8lx", &time_stamp);
        if ( time_stamp >= *time_end )
          {
            if ( fseek (tmp_point[*chan], -rec_size, 1) != 0 )
              {
                fprintf (stderr, "Could not fseek on temp file.\n");
                exit (1);
              }
            *status = 2;
            return;
          }

/* Set up record header. */
        sscanf (record + 4, "%4hx", &header->channel);
        header->time = time_stamp;
        sscanf (record + 16, "%4hx%4hx", &header->year, &header->day);
        for ( n = 24; n < 40; n += 2 )
          {
            sscanf (record + n, "%2hx", &character);
            if ( character != 0x0a )
              *(header->name + (n - 24) / 2) = character;
          }

/* Place the data in the record. */
        if ( *chan == 0 )
          {
            sscanf (record + 40, "%4hx", &console.type);
            for ( n = 44; n < 208; n += 2 )
              {
                sscanf (record + n, "%2hx", &character);
                if ( character != BS )
                  *ascii_pointer++ = character;
                else if ( character == BS )
                  --ascii_pointer;
              }
            *ascii_pointer = '\0';
            list (time_stamp);
          }
        else if ( *chan > 0 && *chan <= 10 )
          {
            sscanf (record + 40, "%4hx", &radar->type);
            for ( n = 44; n < 104; n += 12 )
              sscanf (record + n, "%4hx%8lx", (radar->no_vehicles+(n-44)/12),
                      (radar->speed_sum+(n-44)/12));
          }
        else if ( *chan >= 11 )
          {
            for ( n = 40; n < 88; n += 8 )
              sscanf (record + n, "%4hx%4hx", (regular->type+(n-40)/8),
                      (regular->sample+(n-40)/8));
          }
      }
}
```

Appendix D

SETD Data Reduction Program

```
C
C
C ***********************************************************************
C *                                                                     *
C *                                                                     *
C *                                                                     *
C *                    SETD DATA REDUCTION PROGRAM                      *
C *                                                                     *
C *                                                                     *
C *                   Written by Michael W. Hlavinka                    *
C *                                                                     *
C *                  The Texas Transportation Institute                 *
C *                                                                     *
C *                  The Texas A&M University System                    *
C *                                                                     *
C *                           Project 22830                             *
C *                                                                     *
C *                       College Station, Texas                        *
C *                                                                     *
C *                         in cooperation with                         *
C *                                                                     *
C *                       Texas State Department                        *
C *                                                                     *
C *                  Highways and Public Transportation                 *
C *                                                                     *
C *                                and                                  *
C *                                                                     *
C *                  Department of Chemical Engineering                 *
C *                                                                     *
C *                         Texas A&M University                        *
C *                                                                     *
C *                                                                     *
C ***********************************************************************
C
C
      IMPLICIT INTEGER*4 (A-Z)
      CHARACTER*12 NAME(64),TMPFIL(64)
      CHARACTER*8 UNITS(64)
      CHARACTER*20 LOCAL(64)
      REAL*4 CALCON(64),OFFSET(64),CCON(3)
      REAL*4 RATIO,LARGE(21),SMALL(21)
      REAL*4 CHANGE(10,6)
      REAL*4 ALLSMP(3,6),RSPD(5),RSAMPL(6)
      INTEGER*2 RTYPE(64),MAXVAL(64),DATE
      INTEGER*2 MINVAL(64),INSGRP(64),RADTYP
      INTEGER*4 ACTIVE(64),SPDSUM(5),UVW(6,3),TZERO
      INTEGER*4 TIME(3),ALLTYP(3,6),STMP(3)
      INTEGER*4 FSADCT(64)
      INTEGER*4 SRATE(64),CALDAT(64,9)
      INTEGER*2 HEAD(10),RADAR(16),REG(12),TYPE(6),SAMPLE(6),NOCARS(5)
      EQUIVALENCE (HEAD(1),STAMP),(RADTYP,RADAR(1)),(RADAR(2),NOCARS)
      EQUIVALENCE (RADAR(7),SPDSUM),(REG(1),TYPE),(REG(7),SAMPLE)
      EQUIVALENCE (HEAD(5),DATE)
      LOGICAL CALIB(64)
      COMMON /COR/ LARGE,SMALL
      COMMON NORECS
      DATA TZERO/000000/
      NORECS=0
      OPEN (5,FILE='/users/hlavinka/balcones/sys.dat',STATUS='OLD')
    1 FORMAT('1')
C
C Read the number of active channels.
C
      READ (5,2) NOCHAN
    2 FORMAT(I2)
C
C Read the channel parameters as given to the BALCONES computer in
```

```
C its software.
C
      READ (5,3) (ACTIVE(I),NAME(I),UNITS(I),LOCAL(I),CALCON(I),
     *OFFSET(I),RTYPE(I),MAXVAL(I),MINVAL(I),INSGRP(I),I=1,NOCHAN)
    3 FORMAT(I2,1X,A12,1X,A8,1X,A20,1X,F6.2,1X,F6.2,1X,I2,1X,I5,1X,I5,1X
     *,I3)
C
C Read how the UVW anemometers are grouped.
C
      READ (5,2) NOUVW
      READ (5,590) ((UVW(J,K),K=1,3),J=1,NOUVW)
  590 FORMAT(3I2)
C
C Read the number of days of data.
C
      READ (5,2) NODAYS
C
C Read the calibration channels and the A/D Counts for full scale on
C each calibration channel (FSADCT).
C
      DO 10 J=1,64
      CALIB(J)=.FALSE.
   10 CONTINUE
      DO 20 J=1,64
      READ (5,15,END=30) CHAN,ADVAL
   15 FORMAT (I2,1X,I4)
      CALIB(CHAN)=.TRUE.
      FSADCT(CHAN)=ADVAL
   20 CONTINUE
   30 CONTINUE
      CLOSE (5)
C
C Open calibration data file.
C
      OPEN (1,FILE='/users/hlavinka/balcones/setaout/calib.setd',
     *STATUS='OLD')
C
C Read the elements of CHANGE.  This information marks changes in
C calibration constants that may have occurred during run due to swapping
C instruments.
C
      OPEN (UNIT=2,FILE='/users/hlavinka/balcones/changes.setd',STATUS='
     *OLD')
      READ (2,55,END=65) ((CHANGE(I,J),J=1,6),I=1,10)
   55 FORMAT(F4.0,01X,F3.0,01X,F6.0,01X,F2.0,01X,F6.2,01X,F6.2)
   65 CLOSE (2)
      SYSCON=1
C
C Begin averaging all channels.
C
      DO 165 J=1,64
      DO 175 K=1,9
      CALDAT(J,K)=0
  175 CONTINUE
  165 CONTINUE
      CALL opener (1)
C
      DO 100 KK=1,NODAYS
C
C Split files into daily temporary files.  (C program)
C
      CALL split (1,EOF)
C
C Determine initial sample rates.
C
      CALL sample_rate (SRATE)
C
```

```
C Determine date of collection period.
C
      CALL get_date (YEAR, DAY, MONTH, DAYOFM, SHYEAR)
      WRITE (6,605) YEAR,DAY
      NORECS=NORECS+1
 605  FORMAT(I4.4,1H.,I3.3)
C
C Begin Averaging.  NOTE:  Time stamps are in seconds since midnight.
C
      DO 140 I=1,NOCHAN
      IF (I .EQ. 1) THEN
C
C Print console messages.
C
 130  CALL pass (0,NORECS,STAT,HEAD,RADAR,REG)
      IF (STAT .EQ. 1) GO TO 130
      CALL TERMIN(0)
      END IF
C
C  Write channel information to file.
C
      IF (RTYPE(I).NE.19) THEN
      WRITE (6,620) TZERO,ACTIVE(I),99,99,RTYPE(I),NAME(I),UNITS(I)
      NORECS=NORECS+1
 620  FORMAT(I6.6,4I3,5X,A12,5X,A8)
      END IF
C
C  Get data from data file.
C
      TSTMP=-1
 145  CALL pass (ACTIVE(I),NORECS,STAT,HEAD,RADAR,REG)
      IF (STAT.EQ.0) GO TO 135
      IF (TSTMP.NE.-1) SRATE(ACTIVE(I))=STAMP-TSTMP
      TSTMP=STAMP
C
C  Check if system configuration has been changed.
C
      IF (RTYPE(I).NE.19) THEN
      IF (YEAR.GE.IFIX(CHANGE(SYSCON,1)) .AND. DAY.GE.IFIX(CHANGE(SYSCON
     *,2)) .AND. STAMP.GE. IFIX(CHANGE(SYSCON,3)) .AND. ACTIVE(I).EQ.IFI
     *X(CHANGE(SYSCON,4))) THEN
      CALCON(I)=CHANGE(SYSCON,5)
      OFFSET(I)=CHANGE(SYSCON,6)
      SYSCON=SYSCON+1
      END IF
C
C  Compute hours, minutes and seconds from time stamp.
C
      HH=STAMP/3600
      MM=(STAMP-HH*3600)/60
      SS=STAMP-HH*3600-MM*60
      END IF
C
C  If channel is a radar, compute radar data.
C
      IF (ACTIVE(I).LE.10) THEN
      DO 150 K=1,5
      IF (NOCARS(K).GT.0) THEN
      RSPD(K)=SPDSUM(K)/409.4*20.*SQRT(2.)/NOCARS(K)
      ELSE
      RSPD(K)=0.
      END IF
 150  CONTINUE
      WRITE (6,600) HH,MM,SS,ACTIVE(I),RADTYP,
     *(NOCARS(K),RSPD(K),K=1,5)
      NORECS=NORECS+1
 600  FORMAT(3I2.2,1X,I2,1X,I2,1X,5(I5,1X,F5.1))
```

```
C
C  If channel is a calibration channel, compute the data.
C
      ELSE IF (CALIB(ACTIVE(I))) THEN
      DO 160 K=1,6
      IF (((IFIX((K-1)*SRATE(ACTIVE(I))/6.+STAMP).GT.CALDAT(ACTIVE(I),4)
     * .AND. DATE.EQ.CALDAT(ACTIVE(I),3)) .OR. DATE.GT.CALDAT(ACTIVE(I),
     *3)) .AND. TYPE(K) .EQ. RTYPE(I)) THEN
 510  READ (1,520,END=530) CHAN,TIME1,YEAR,DAY1,DAY2,TIME2,SPAN1,SPAN2,Z
     *ERO1,ZERO2
 520  FORMAT(Z4,Z8,Z4,Z4,Z4,Z8,Z4,Z4,Z4,Z4)
      DTIME=(DAY2-DAY1-1)*86400+(86400-TIME1)+TIME2
      CALDAT(CHAN,1)=DAY1
      CALDAT(CHAN,2)=TIME1
      CALDAT(CHAN,3)=DAY2
      CALDAT(CHAN,4)=TIME2
      CALDAT(CHAN,5)=DTIME
      CALDAT(CHAN,6)=SPAN1
      CALDAT(CHAN,7)=SPAN2
      CALDAT(CHAN,8)=ZERO1
      CALDAT(CHAN,9)=ZERO2
      IF (CHAN.NE.ACTIVE(I)) GO TO 510
      END IF
 530  IF (TYPE(K).EQ.RTYPE(I)) THEN
      ZDRIFT=FLOAT(CALDAT(ACTIVE(I),9)-CALDAT(ACTIVE(I),8))/FLOAT(CALDAT
     *(ACTIVE(I),5))*((K-1)*SRATE(ACTIVE(I))/6.+STAMP-CALDAT(ACTIVE(I),2
     *))
      SDRIFT=FLOAT(CALDAT(ACTIVE(I),7)-CALDAT(ACTIVE(I),6))/FLOAT(CALDAT
     *(ACTIVE(I),5))*((K-1)*SRATE(ACTIVE(I))/6.+STAMP-CALDAT(ACTIVE(I),2
     *))+CALDAT(ACTIVE(I),6)-ZDRIFT
      RSAMPL(K)=(SAMPLE(K)-ZDRIFT)*(1.+FLOAT(CALDAT(ACTIVE(I),6)-SDRIFT)
     */FLOAT(FSADCT(ACTIVE(I))))
      ELSE
      RSAMPL(K)=SAMPLE(K)
      END IF
      RSAMPL(K)=RSAMPL(K)/409.4*CALCON(I)+OFFSET(I)
 160  CONTINUE
C
C  Write the data.
C
      IF (RTYPE(I).NE.4) THEN
      WRITE (6,610) HH,MM,SS,ACTIVE(I),(TYPE(K),RSAMPL(K),K=1,6)
      NORECS=NORECS+1
      ELSE
      WRITE (6,615) HH,MM,SS,ACTIVE(I),(TYPE(K),RSAMPL(K),K=1,6)
      NORECS=NORECS+1
      END IF
C
C  Process data for UVW anemometers
C
      ELSE IF (RTYPE(I).EQ.19) THEN
C
C  Determine grouping of UVW anemometers
C
      DO 400 ANEM=1,NOUVW
      IF (ACTIVE(I).EQ.UVW(ANEM,1)) GO TO 410
C
C   Go to 470 if UVW corrections have already been determined.
C
      IF (ACTIVE(I).EQ.UVW(ANEM,2)) GO TO 470
      IF (ACTIVE(I).EQ.UVW(ANEM,3)) GO TO 470
 400  CONTINUE
 410  CONTINUE
C
C  Write channel information to file.
C
      WRITE (6,620) TZERO,ACTIVE(I),UVW(ANEM,2),UVW(ANEM,3),RTYPE(I),
```

```
     *NAME(I),UNITS(I)
      NORECS=NORECS+1
C
C  Determine calibration constants for UVW anemometers in group
C
      CCON(1)=CALCON(I)
      DO 420 J=I,NOCHAN
      IF (ACTIVE(J).EQ.UVW(ANEM,2)) CCON(2)=CALCON(J)
      IF (ACTIVE(J).EQ.UVW(ANEM,3)) CCON(3)=CALCON(J)
 420  CONTINUE
C
C  Read UVW data for two channels that data has not been read for yet.
C
      STMP(1)=STAMP
      TIME(1)=STAMP
      STAT1=STAT
      DO 440 K=1,6
      ALLTYP(1,K)=TYPE(K)
      ALLSMP(1,K)=SAMPLE(K)
 440  CONTINUE
      CALL pass (UVW(ANEM,2),NORECS,STAT2,HEAD,RADAR,REG)
      DO 441 K=1,6
      ALLTYP(2,K)=TYPE(K)
      ALLSMP(2,K)=SAMPLE(K)
 441  CONTINUE
      STMP(2)=STAMP
      TIME(2)=STAMP
      CALL pass (UVW(ANEM,3),NORECS,STAT3,HEAD,RADAR,REG)
      DO 442 K=1,6
      ALLTYP(3,K)=TYPE(K)
      ALLSMP(3,K)=SAMPLE(K)
 442  CONTINUE
      STMP(3)=STAMP
      TIME(3)=STAMP
C
C  Open temporary files to hold UVW data.
C
      WRITE (TMPFIL(UVW(ANEM,2)),630) UVW(ANEM,2)
      WRITE (TMPFIL(UVW(ANEM,3)),630) UVW(ANEM,3)
 630  FORMAT('TMP',I2.2)
      OPEN (8,FILE=TMPFIL(UVW(ANEM,2)),STATUS='NEW')
      OPEN (9,FILE=TMPFIL(UVW(ANEM,3)),STATUS='NEW')
C
 430  CONTINUE
C
C  Get samples at TIME and apply UVW corrections.  Write channel
C  ACTIVE(I) to unit 6 and the other 2 anemometers to the temporary
C  files.
C
 450  K1=(TIME(1)-STMP(1))/(SRATE(UVW(ANEM,1))/6.)+1
      K2=(TIME(2)-STMP(2))/(SRATE(UVW(ANEM,2))/6.)+1
      K3=(TIME(3)-STMP(3))/(SRATE(UVW(ANEM,3))/6.)+1
C
C  Check if system configuration has changed.
C
      IF (YEAR.GE.IFIX(CHANGE(SYSCON,1)) .AND. DAY.GE.IFIX(CHANGE(SYSCON
     *,2)) .AND. TIME(1).GE.IFIX(CHANGE(SYSCON,3)) .AND. UVW(ANEM,1).EQ.
     *IFIX(CHANGE(SYSCON,4))) THEN
      CALCON(I)=CHANGE(SYSCON,5)
      OFFSET(I)=CHANGE(SYSCON,6)
      CCON(I)=CALCON(I)
      SYSCON=SYSCON+1
      END IF
      IF (YEAR.GE.IFIX(CHANGE(SYSCON,1)) .AND. DAY.GE.IFIX(CHANGE(SYSCON
     *,2)) .AND. TIME(2).GE.IFIX(CHANGE(SYSCON,3)) .AND. UVW(ANEM,2).EQ.
     *IFIX(CHANGE(SYSCON,4))) THEN
      CALCON(J)=CHANGE(SYSCON,5)
```

```
      OFFSET(J)=CHANGE(SYSCON,6)
      CCON(2)=CALCON(J)
      SYSCON=SYSCON+1
      END IF
      IF (YEAR.GE.IFIX(CHANGE(SYSCON,1)) .AND. DAY.GE.IFIX(CHANGE(SYSCON
     *,2)) .AND. TIME(3).GE.IFIX(CHANGE(SYSCON,3)) .AND. UVW(ANEM,3).EQ.
     *IFIX(CHANGE(SYSCON,4))) THEN
      CALCON(J)=CHANGE(SYSCON,5)
      OFFSET(J)=CHANGE(SYSCON,6)
      CCON(3)=CALCON(J)
      SYSCON=SYSCON+1
      END IF
C
C  Determine if more UVW data needs reading.
C
      IF (K1.GT.6) THEN
      HH=STMP(1)/3600
      MM=(STMP(1)-HH*3600)/60
      SS=STMP(1)-HH*3600-MM*60
      WRITE (6,610) HH,MM,SS,UVW(ANEM,1),(ALLTYP(1,K),ALLSMP(1,K),
     *K=1,6)
      NORECS=NORECS+1
      CALL pass (UVW(ANEM,1),NORECS,STAT1,HEAD,RADAR,REG)
      SRATE(UVW(ANEM,1))=STAMP-STMP(1)
      STMP(1)=STAMP
      DO 460 K=1,6
      ALLTYP(1,K)=TYPE(K)
      ALLSMP(1,K)=SAMPLE(K)
 460  CONTINUE
      K1=1
      END IF
      IF (K2.GT.6) THEN
      HH=STMP(2)/3600
      MM=(STMP(2)-HH*3600)/60
      SS=STMP(2)-HH*3600-MM*60
      WRITE (8,610) HH,MM,SS,UVW(ANEM,2),(ALLTYP(2,K),ALLSMP(2,K),
     *K=1,6)
      CALL pass (UVW(ANEM,2),NORECS,STAT2,HEAD,RADAR,REG)
      SRATE(UVW(ANEM,2))=STAMP-STMP(2)
      STMP(2)=STAMP
      DO 461 K=1,6
      ALLTYP(2,K)=TYPE(K)
      ALLSMP(2,K)=SAMPLE(K)
 461  CONTINUE
      K2=1
      END IF
      IF (K3.GT.6) THEN
      HH=STMP(3)/3600
      MM=(STMP(3)-HH*3600)/60
      SS=STMP(3)-HH*3600-MM*60
      WRITE (9,610) HH,MM,SS,UVW(ANEM,3),(ALLTYP(3,K),ALLSMP(3,K),
     *K=1,6)
      CALL pass (UVW(ANEM,3),NORECS,STAT3,HEAD,RADAR,REG)
      SRATE(UVW(ANEM,3))=STAMP-STMP(3)
      STMP(3)=STAMP
      DO 462 K=1,6
      ALLTYP(3,K)=TYPE(K)
      ALLSMP(3,K)=SAMPLE(K)
 462  CONTINUE
      K3=1
      END IF
C
      IF (TIME(1).EQ.TIME(2).AND.TIME(1).EQ.TIME(3).AND.TIME(2).EQ.TIME(
     *3)) THEN
C
C  Determine response factors
C
```

```
      IF (ALLTYP(1,K1).EQ.19.AND.ALLTYP(2,K2).EQ.19.AND.ALLTYP(3,K3).EQ.
     *19) THEN
      IF (ALLSMP(1,K1).EQ.0..OR.ALLSMP(2,K2).EQ.0.) THEN
      RATIO=0.
      ELSE
      RATIO=ABS(ALLSMP(1,K1)/ALLSMP(2,K2))
      IF (RATIO.GT.1.) RATIO=1./RATIO
      END IF
C
C  Apply response factors
C
      INDEX=IFIX((RATIO*100+2)/5)+1
      IF (ABS(ALLSMP(1,K1)).GE.ABS(ALLSMP(2,K2))) THEN
      ALLSMP(1,K1)=ALLSMP(1,K1)*LARGE(INDEX)
      ALLSMP(2,K2)=ALLSMP(2,K2)*SMALL(INDEX)
      ELSE
      ALLSMP(1,K1)=ALLSMP(1,K1)*SMALL(INDEX)
      ALLSMP(2,K2)=ALLSMP(2,K2)*LARGE(INDEX)
      END IF
      END IF
      ALLSMP(1,K1)=ALLSMP(1,K1)*CCON(1)/409.4
      ALLSMP(2,K2)=ALLSMP(2,K2)*CCON(2)/409.4
      ALLSMP(3,K3)=ALLSMP(3,K3)*CCON(3)/409.4
      TIME(1)=TIME(1)+SRATE(UVW(ANEM,1))/6.
      TIME(2)=TIME(2)+SRATE(UVW(ANEM,2))/6.
      TIME(3)=TIME(3)+SRATE(UVW(ANEM,3))/6.
C
      ELSE IF (TIME(1).LT.TIME(2).OR.TIME(1).LT.TIME(3)) THEN
      ALLSMP(1,K1)=ALLSMP(1,K1)*CCON(1)/409.4
      TIME(1)=TIME(1)+SRATE(UVW(ANEM,1))/6.
      ELSE IF (TIME(2).LT.TIME(1).OR.TIME(2).LT.TIME(3)) THEN
      ALLSMP(2,K2)=ALLSMP(2,K2)*CCON(2)/409.4
      TIME(2)=TIME(2)+SRATE(UVW(ANEM,2))/6.
      ELSE IF (TIME(3).LT.TIME(1).OR.TIME(3).LT.TIME(2)) THEN
      ALLSMP(3,K3)=ALLSMP(3,K3)*CCON(3)/409.4
      TIME(3)=TIME(3)+SRATE(UVW(ANEM,3))/6.
      END IF
      IF (STAT1.EQ.0.AND.STAT2.EQ.0.AND.STAT3.EQ.0) GO TO 480
      GO TO 450
 470  CONTINUE
C
C  This part of the program simply reads the data from previously
C  calculated anemometers and writes it to unit 6.
C
      IF (ACTIVE(I).EQ.UVW(ANEM,2)) THEN
C
C  Write channel information to file.
C
      WRITE (6,620) TZERO,ACTIVE(I),UVW(ANEM,1),UVW(ANEM,3),RTYPE(I),
     *NAME(I),UNITS(I)
      NORECS=NORECS+1
      OPEN (8,FILE=TMPFIL(ACTIVE(I)))
 472  READ (8,610,END=471) HH,MM,SS,CHNO,(TYPE(K),RSAMPL(K),K=1,6)
      WRITE (6,610) HH,MM,SS,CHNO,(TYPE(K),RSAMPL(K),K=1,6)
      NORECS=NORECS+1
      GO TO 472
 471  CONTINUE
      CLOSE(8,STATUS='DELETE')
      ELSE IF (ACTIVE(I).EQ.UVW(ANEM,3)) THEN
C
C  Write channel information to file.
C
      WRITE (6,620) TZERO,ACTIVE(I),UVW(ANEM,1),UVW(ANEM,2),RTYPE(I),
     *NAME(I),UNITS(I)
      NORECS=NORECS+1
      OPEN (9,FILE=TMPFIL(ACTIVE(I)))
 474  READ (9,610,END=475) HH,MM,SS,CHNO,(TYPE(K),RSAMPL(K),K=1,6)
```

```
      WRITE (6,610) HH,MM,SS,CHNO,(TYPE(K),RSAMPL(K),K=1,6)
      NORECS=NORECS+1
      GO TO 474
 475  CONTINUE
      CLOSE(9,STATUS='DELETE')
      END IF
      GO TO 490
 480  CLOSE (8)
      CLOSE (9)
C
C  Rewind temporary UVW split files so that STAT is not zero when
C  checked at top of DO 140 loop
C
      CALL rwnd (UVW(ANEM,2))
      CALL rwnd (UVW(ANEM,3))
 490  GO TO 135
C
C  If channel is not a radar, calibration channel, or UVW anemometer,
C  calculate the data.
C
      ELSE
      DO 300 K=1,6
      RSAMPL(K)=SAMPLE(K)/409.4*CALCON(I)+OFFSET(I)
 300  CONTINUE
      IF (RTYPE(I).NE.3) THEN
      WRITE (6,610) HH,MM,SS,ACTIVE(I),(TYPE(K),RSAMPL(K),K=1,6)
      NORECS=NORECS+1
      ELSE
      WRITE (6,615) HH,MM,SS,ACTIVE(I),(TYPE(K),RSAMPL(K),K=1,6)
      NORECS=NORECS+1
      END IF
      END IF
C
      GO TO 145
C
C  Close temporary file for the channel.
C
 135  CALL closer (ACTIVE(I))
      CALL TERMIN(ACTIVE(I))
 140  CONTINUE
      CALL TERMIN(99)
C
C  Close any temporary files that may remain open.
C
      CALL split (0,EOF)
 610  FORMAT(3I2.2,1X,I2,6(I3,1X,F7.2))
 615  FORMAT(3I2.2,1X,I2,6(I3,1X,F7.3))
 100  CONTINUE
      CALL opener (2)
      WRITE (7,710) NORECS
 710  FORMAT(' ','Total number of records written = ',I8)
      STOP
      END
C
C This subroutine is called to write terminator records at end of
C channel or end of day.
C
      SUBROUTINE TERMIN(ACTIVE)
      INTEGER TYP(6),TIME,ACTIVE
      REAL SAMP(6),RADSMP(5)
      DATA TYP/6*99/,TIME/999999/
      DATA SAMP/6*9999.99/,RADSMP/5*999.9/
      IF (ACTIVE.LE.10.AND.ACTIVE.NE.0) THEN
      WRITE (6,600) TIME,ACTIVE,TYP(1),(TYP(K),RADSMP(K),K=1,5)
      NORECS=NORECS+1
 600  FORMAT(I6.6,1X,I2,1X,I2,1X,5(I5,1X,F5.1))
      ELSE
```

```
      WRITE (6,610) TIME,ACTIVE,(TYP(K),SAMP(K),K=1,6)
      NORECS=NORECS+1
  610 FORMAT(I6.6,1X,I2,6(I3,1X,F7.2))
      END IF
      RETURN
      END
C
C
C This BLOCK DATA subprogram assigns all of the correction factors
C for UVW non-cosine response.
C
      BLOCK DATA
      COMMON /COR/ LARGE,SMALL
      REAL*4 LARGE(21),SMALL(21)
      DATA LARGE /1.,1.,1.,1.002,1.004,1.008,1.013,1.018,1.024,1.03,1.03
     *7,1.043,1.049,1.057,1.066,1.075,1.083,1.093,1.103,1.115,1.130/
      DATA SMALL /1.25,1.25,1.25,1.25,1.25,1.25,1.248,1.245,1.240,1.235,
     *1.228,1.22,1.212,1.203,1.193,1.183,1.173,1.163,1.152,1.141,1.130/
      END
```

```
/* These are all of the C programs used in conjunction with the FORTRAN
   SETD program. */


#include <stdio.h>
#include <errno.h>
extern int errinfo, errno;

FILE *tmpfile (), *tmp_point[65], *fopen (), *input;

#define         NO_CHAN         64              /* Total number of possible
                                                   channels on A/D */

unsigned char record[220];

/* This structure defines the beginning for each record.  It is the same
   on all records. */
struct general_info
 {
   long int time;
   short int channel;
   short int year;
   short int day;
   char name[10];
 };

/* This structure defines the data in the radar records. */
struct rad_params
 {
   short int type;
   short int no_vehicles[5];
   long int speed_sum[5];
 };

/* This structure defines the data in the regular channel records. */
struct reg_params
 {
   short int type[6];
   short int sample[6];
 };

/* This structure defines the data in the console messages. */
struct
 {
   short type;
   unsigned char ASCII[82];
 } console;

/* This routine will skip to a specified day if desired. */
skipper (day, offset)
  int *day;
  long *offset;
 {
   unsigned char data[220];
   int read_day, length;
   fseek (input, *offset, 0);
   fgets (data, 220, input);
   while ( fgets (data, 220, input) != NULL )
    {
      sscanf (data + 20, "%4x", &read_day);
      if ( read_day == *day )
        {
          sscanf (data, "%4x", &length);
          length *= 2;
          ++length;
          if ( fseek (input, -length, 1) != 0 )
            {
```

```
                fprintf (stderr, "Could not fseek in skipper.\n");
                exit (1);
              }
            return;
          }
        else if ( read_day > *day )
          {
            fprintf (stderr, "Offset to large.  Desired date passed.\n");
            exit (-1);
          }
      }
    fprintf (stderr, "Illegal offset for skipper.  End of file encountered.\n");
    exit (1);
  }

/* This routine opens the raw data file. */
opener (option)
 int *option;
 {
    if (*option == 1)
     {
      if ( (input = fopen ("/users/hlavinka/balcones/setaout/setb.out",
                           "r")) == NULL )
        {
          fprintf (stderr, "Input data could not be opened successfully.\n");
          fprintf (stderr, "errinfo = %d; errno = %d\n", errinfo, errno);
          exit (-1);
        }
     }
    else if (*option == 2)
      {
        fclose (input);
        input = NULL;
      }
 }


/* This routine splits the large data file into small temporary files for
   each channel for each day.  After all averages have been calculated for
   the day, the files are closed and thus deleted. */

split (request, end_of_file)
  int *request;   /* This determines if files are split or closed. */
  int *end_of_file;  /* Determines when at EOF in raw data file. */
 {
   short int channel;
   long int date, old_date, length;
   int n = 0;
   char *get_record;

/* If request = 1 then split file. */
   if ( *request )
     {
      *end_of_file = 0;

/* Null all temporary file pointers. */
      for ( n = 0; n <= NO_CHAN; ++n )
        tmp_point[n] = NULL;
      fgets (record, 220, input);
      sscanf (record + 4, "%4hx", &channel);
      sscanf (record + 16, "%8lx", &old_date);
      if ( (tmp_point[channel] = tmpfile () ) == NULL )
        {
          fprintf (stderr, "Console message temp file not opened.\n");
          fprintf (stderr, "errinfo = %d; errno = %d\n", errinfo, errno);
          exit (-1);
        }
```

```
      while ( (get_record = fgets (record, 220, input) ) != NULL )
        {
          sscanf (record, "%4x", &length);
          sscanf (record + 4, "%4hx", &channel);
          sscanf (record + 16, "%8lx", &date);
          if ( tmp_point[channel] == NULL )
            {
              if ( (tmp_point[channel] = tmpfile () ) == NULL &&
                      channel <= NO_CHAN )
                {
                  fprintf (stderr, "Temp file not opened.  Channel = %d\n",
                           channel);
                  fprintf (stderr, "errinfo = %d; errno = %d\n",
                                           errinfo, errno);
                  exit (-1);
                }
            }
          if ( date == old_date && channel <= NO_CHAN )
            fputs (record, tmp_point[channel]);
          else if ( date != old_date )
            break;
        }
      length *= 2;
      ++length;
      if ( fseek (input, -length, 1) != 0 )
        {
          fprintf (stderr, "fseek could not backup a record.\n");
          fprintf (stderr, "errinfo = %d; errno = %d\n", errinfo, errno);
          exit (-1);
        }
      for ( n = 0; n <= NO_CHAN; ++n )
        {
          if ( tmp_point[n] != NULL )
            rewind (tmp_point[n]);
        }
      if ( get_record == NULL )
        *end_of_file = 1;

    }
  else
    {
      for ( n = 0; n <= NO_CHAN; ++n )
        {
          if ( tmp_point[n] != NULL )
            {
              while ( fclose (tmp_point[n]) == EOF )
                {
                  fprintf (stderr, "Could not close a file properly n = %d.\n",
                                   n );
                  fprintf (stderr, "errinfo = %d; errno = %d\n",
                                                 errinfo, errno);
                }
              tmp_point[n] = NULL;
            }
        }
    }
}


/* This routine determines the sample rates of each channel.  The returned
   information is the time between subsequent records for all channels. */
sample_rate (srate)
 int *srate;
 {
   int n, time1, time2, rec_size;
   for ( n = 1; n <= NO_CHAN; ++n )
     {
```

```
      if ( n <= 10 )
        rec_size = -105;
      else if ( n >= 11 )
        rec_size = -89;
      if ( tmp_point[n] != NULL )
        {
          if (fgets (record, 220, tmp_point[n]) != NULL)
            {
              sscanf (record + 8, "%8lx", &time1);
              if (fgets (record, 220, tmp_point[n]) != NULL)
                {
                  sscanf (record + 8, "%8lx", &time2);
                  *(srate + n - 1) = time2 - time1;
                  if ( fseek (tmp_point[n], 2 * rec_size, 1) != 0 )
                    {
                      fprintf (stderr, "Could not fseek in srate.\n");
                      fprintf (stderr, "errinfo = %d; errno = %d\n",
                                                 errinfo, errno);
                      exit (-1);
                    }
                }
              else
                {
                  if ( fseek(tmp_point[n], rec_size, 1) != 0 )
                    {
                      fprintf (stderr, "Could not fseek in srate.\n");
                      fprintf (stderr, "errinfo = %d; errno = %d\n",
                                                 errinfo, errno);
                      exit (-1);
                    }
                }
            }
        }
      else
        *(srate + n - 1) = 0;
    }
}


/* This routine gets the date for each run.  It is called directly
   from FORTRAN before any data for the day has been processed. */
get_date (year, date, month, day_of_month, two_digit_year)
 int *year, *date, *month, *day_of_month, *two_digit_year;
 {
   char *convert ();
   struct general_info header;
   fgets (record, 220, tmp_point[0]);

/* Get date out of first console record. */
   sscanf (record + 16, "%4hx%4hx", &header.year, &header.day);
   rewind (tmp_point[0]);
   *year = header.year;
   *date = header.day;
 }


/* This routine lists ASCII console messages to standard out. */
list (time_stamp)
 long int time_stamp;
 {
   unsigned char *eol;
   struct
    {
      short int hour;
      short int minute;
      short int second;
```

```
    } time;
      time.hour = time_stamp / 3600;
      time.minute = (time_stamp - time.hour * 3600) / 60;
      time.second = time_stamp - time.hour * 3600 - time.minute * 60;

/* Strip trailing spaces from console message. */
      eol = console.ASCII + strlen (console.ASCII) - 1;
      while ( *eol == 0x20 )
        *eol-- = '\0';

      fprintf (stdout, "%02d%02d%02d 00 %.70s\n", time.hour, time.minute,
          time.second, console.ASCII);
      if (strlen (console.ASCII) > 70)
        fprintf (stdout, "%02d%02d%02d 00 %s\n", time.hour, time.minute,
            time.second, console.ASCII + 70);
      fflush (stdout);
 }


/* This routine passes data back to the FORTRAN program.  The FORTRAN program
   then does all the calculations. */
pass (chan, norecs, status, header, radar, regular)
 long int *chan, *norecs, *status;
 struct general_info *header;
 struct reg_params *regular;
 struct rad_params *radar;
 {
   long int rec_size, time_stamp;
   short unsigned int character;
   int n;
   unsigned char BS = 0x08;
   unsigned char *ascii_pointer = console.ASCII;

/* Set up rec_size for hex ascii records. */
   if ( *chan >= 11 )
     rec_size = 89;      /* Regular records */
   else if ( *chan <= 10 && *chan > 0 )
     rec_size = 105;     /* Radar records */
   else if ( *chan == 0 )
     rec_size = 209;     /* Console messages */

/* Read records from time begin to time end. */
   if ( tmp_point[*chan] == NULL )
     {
       *status = 0;
       return;
     }
   if ( fgets (record, 220, tmp_point[*chan]) == NULL )
     *status = 0;
   else
     {
       *status = 1;
       sscanf (record + 8, "%8lx", &time_stamp);

/* Set up record header. */
       sscanf (record + 4, "%4hx", &header->channel);
       header->time = time_stamp;
       sscanf (record + 16, "%4hx%4hx", &header->year, &header->day);
       for ( n = 24; n < 40; n += 2 )
         {
           sscanf (record + n, "%2hx", &character);
           if ( character != 0x0a )
             *(header->name + (n - 24) / 2) = character;
         }

/* Place the data in the record. */
       if ( *chan == 0 )
```

```
        {
          sscanf (record + 40, "%4hx", &console.type);
          for ( n = 44; n < 208; n += 2 )
            {
              sscanf (record + n, "%2hx", &character);
              if ( character != BS )
                *ascii_pointer++ = character;
              else if ( character == BS )
                --ascii_pointer;
            }
          *ascii_pointer = '\0';
          (*norecs)++;
          list (time_stamp);
        }
      else if ( *chan > 0 && *chan <= 10 )
        {
          sscanf (record + 40, "%4hx", &radar->type);
          for ( n = 44; n < 104; n += 12 )
            sscanf (record + n, "%4hx%8lx", (radar->no_vehicles+(n-44)/12),
                    (radar->speed_sum+(n-44)/12));
        }
      else if ( *chan >= 11 )
        {
          for ( n = 40; n < 88; n += 8 )
            sscanf (record + n, "%4hx%4hx", (regular->type+(n-40)/8),
                    (regular->sample+(n-40)/8));
        }
    }
}

/* This routine closes the file associated with chan. */
closer (chan)
 long int *chan;
{
  int try = 0;
  if ( tmp_point[*chan] != NULL )
    {
      while ( fclose (tmp_point[*chan]) == EOF )
        {
          fprintf (stderr, "Could not close a file in closer n = %d.\n",
                                              *chan);
          ++try;
          fprintf (stderr, "errinfo = %d; errno = %d\n", errinfo, errno);
          if ( try > 1000 )
            exit (-1);
        }
    }
  tmp_point[*chan] = NULL;
}

/* This routine rewinds the file associated with chan. */
rwnd (chan)
 long int *chan;
{
  rewind (tmp_point[*chan]);
}
```

Appendix E

Radian Calibration Confirmation Report

VERIFICATION OF SF_6 CONCENTRATION IN MATHESON CALIBRATION GAS CYLINDER

Verification of the SF_6 concentration in the gas cylinder submitted for analysis by Texas A & M was accomplished by using two separate sources of standard gas from two different vendors. A cylinder of 105 ppb SF_6 standard prepared in air was diluted to 1,2, and 4 ppb using an all stainless steel capillary dilution device designed by Radian Corporation personnel. This device prepares gas mixtures dynamically such that the mixture is never contained for any period of time thereby eliminating the permeation or condensation problems encountered in static systems.

The second calibration cylinder was obtained at a concentration of 2.02 ppb SF_6, with nitrogen used as the diluent. This was the expected concentration of the Texas A & M standard gas cylinder.

The procedure for analysis was the same for all SF_6 sources. Gas from gas cylinders or the capillary dilution device was passed through a 2 cc stainless steel sample loop. After a thorough 10-second flush of the loop and equilibration to atmospheric pressure, the gas chromatograph carrier gas was diverted to flush the contents of the loop onto the GC column by means of a 10-port Valco valve. The column and conditions for GC analysis are as follows:

Tracor 560 GC

Hewlett Packard 3380A Integrator

Column: 6' x 4 mm I.D. glass packed with 1.5% XE-60/1% H_3PO_4 on Carbopack B

Column Temperature: 50°C

Detector Temperature: 310°C

Injector Temperature: 200°C

Carrier: 5% Methane/95% Argon at 20 mL/min

SF_6 Retention Time: 2.0 minutes

Results of the analyses are presented in the attached table. The SF_6 concentration of the Texas A & M cylinder was calculated relative to the 2.02 ppb source and the 2 ppb dilution prepared from the 105 ppb source gas. The Texas A & M cylinder was determined to be 2.05 and 2.15 ppb SF_6 from the respective analyses.

RESULTS OF SF_6 VERIFICATION STUDY

(RC #225-062)

Source	Area Counts Mean ± S.D.	Number of Replicates	SF_6 Concentration (ppb)	
			Relative to 2.02 ppb Standard	Relative to 2 ppb Diluted from 105 ppb Standard
105 ppb SF_6 Cylinder	289,603 ± 2,051	3	113	118
2.02 ppb SF_6 Cylinder	5,183 ± 47	8	2.02	2.14
1 ppb SF_6 Dilution*	2,256 ± 71	4	0.88	0.92
2 ppb SF_6 Dilution*	4,899 ± 97	5	1.91	2.0
4 ppb SF_6 Dilution*	11,287 ± 594	6	4.40	4.61
Texas A & M Cylinder	5,256 ± 25	5	2.05	2.15

*Dilution from 105 ppb SF_6 bottle, using dynamic dilution device.

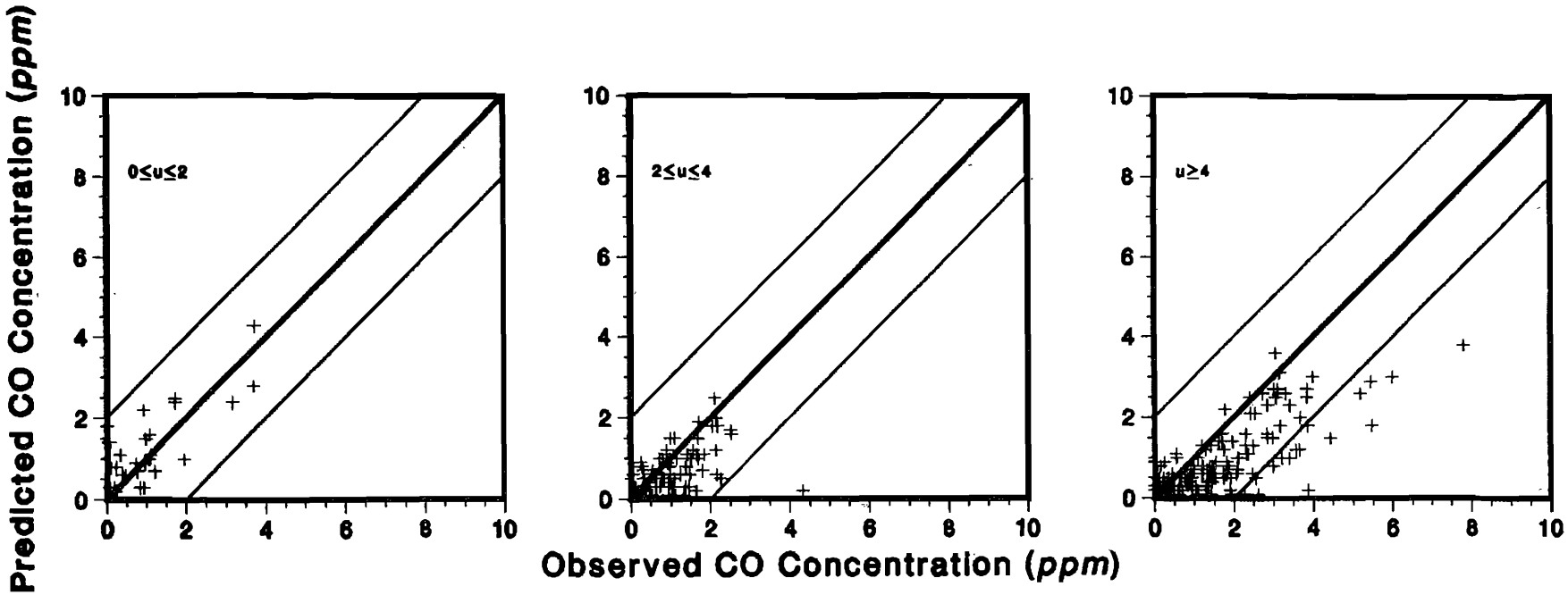
Appendix F

Scatterplots for Detailed Analysis of TEXIN2

This appendix contains the individual scatterplots for the various wind speed/angle combinations discussed in Chapter 7. Scatterplots for each receptor in the College Station data base are also presented. For the receptor scatterplots, refer to Figure 21 of Chapter 2 to determine the tower locations.

Scatterplots for Detailed Analysis of TEXIN2

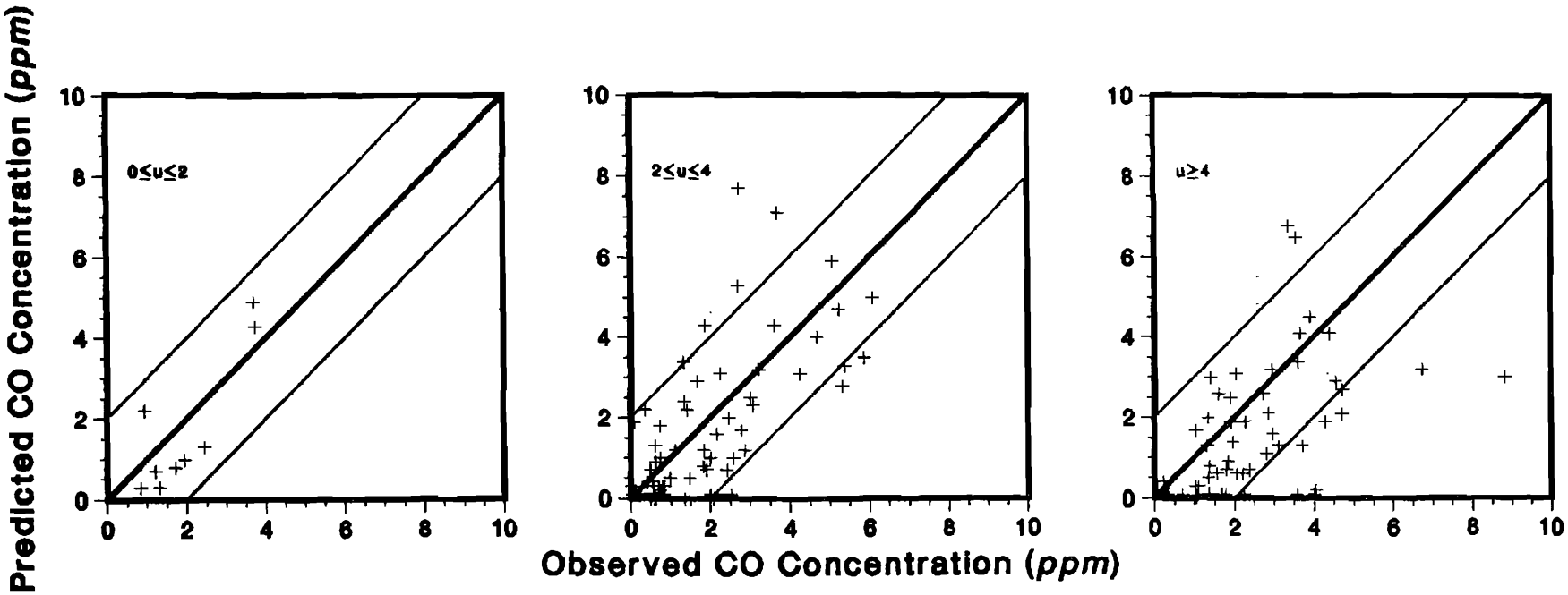
Scatterplots for Detailed Analysis of TEXIN2 (Continued)



Scatterplots for Near Parallel Wind Cases

Original TEXIN Model

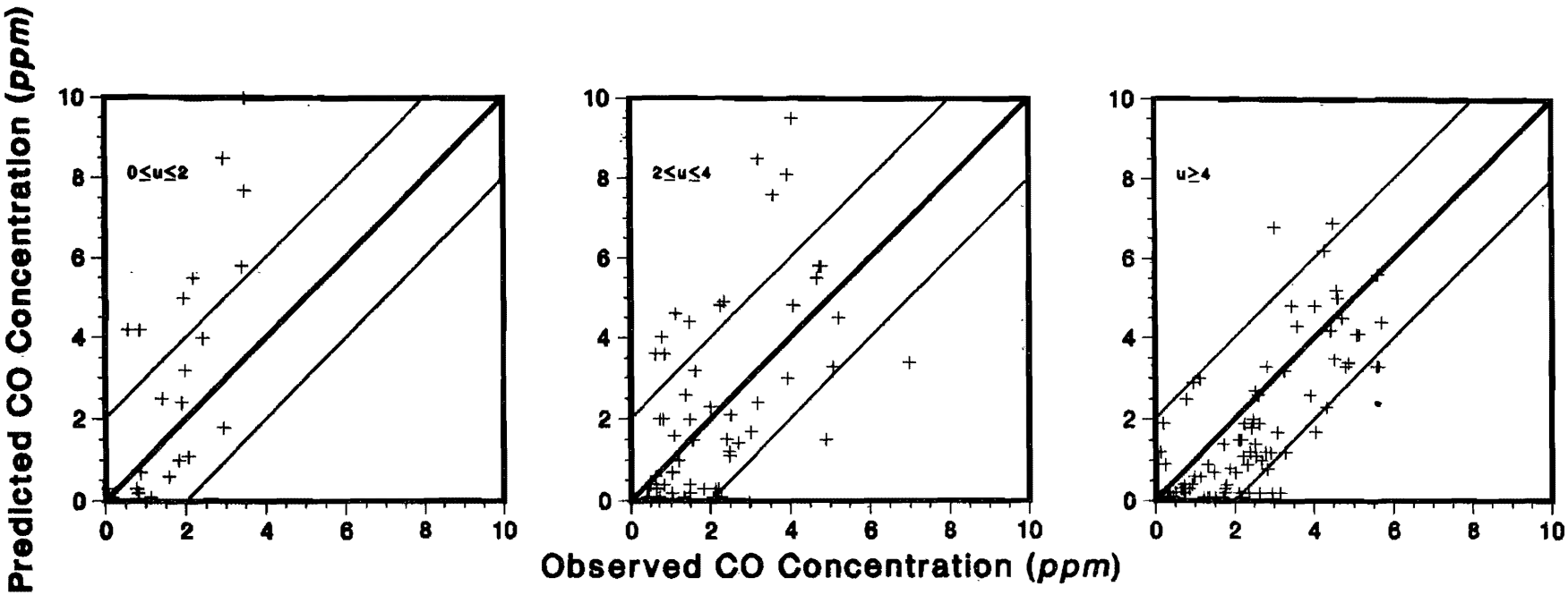
College Station Data Base



Scatterplots for Near 45° Wind Cases

Original TEXIN Model

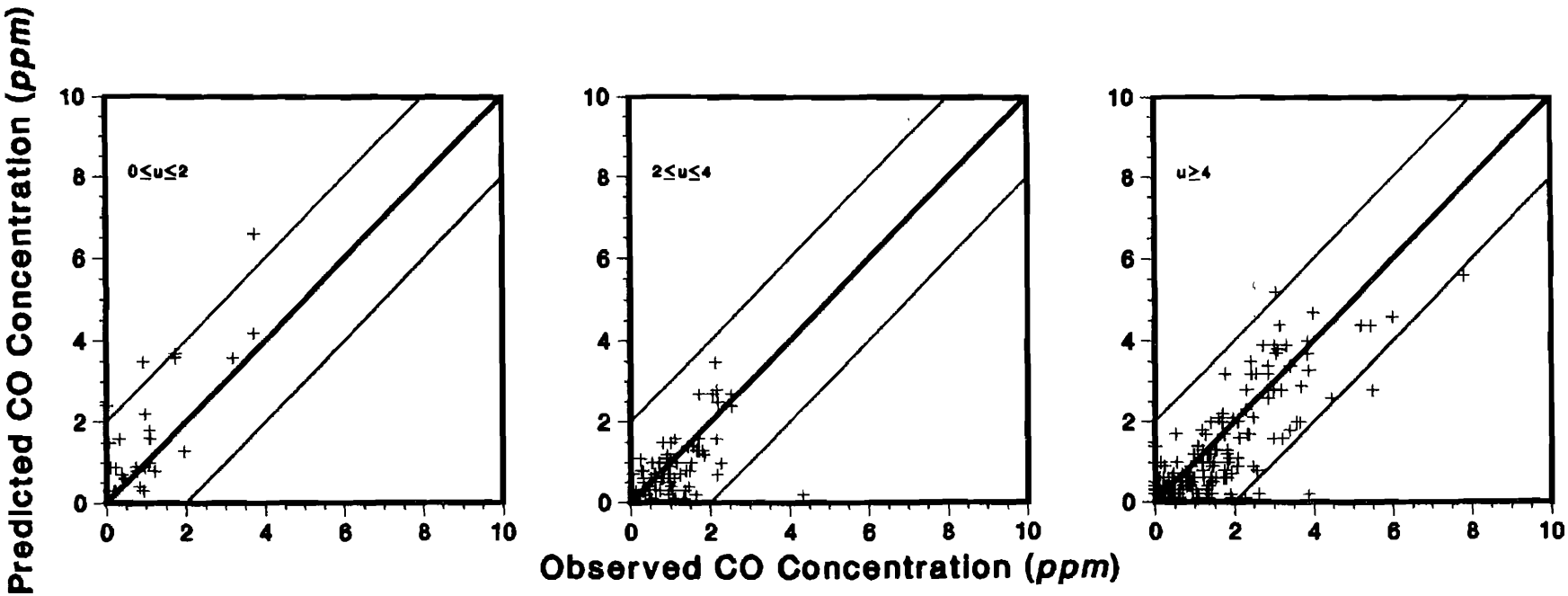
College Station Data Base



Scatterplots for Near Perpendicular Wind Cases

Original TEXIN Model

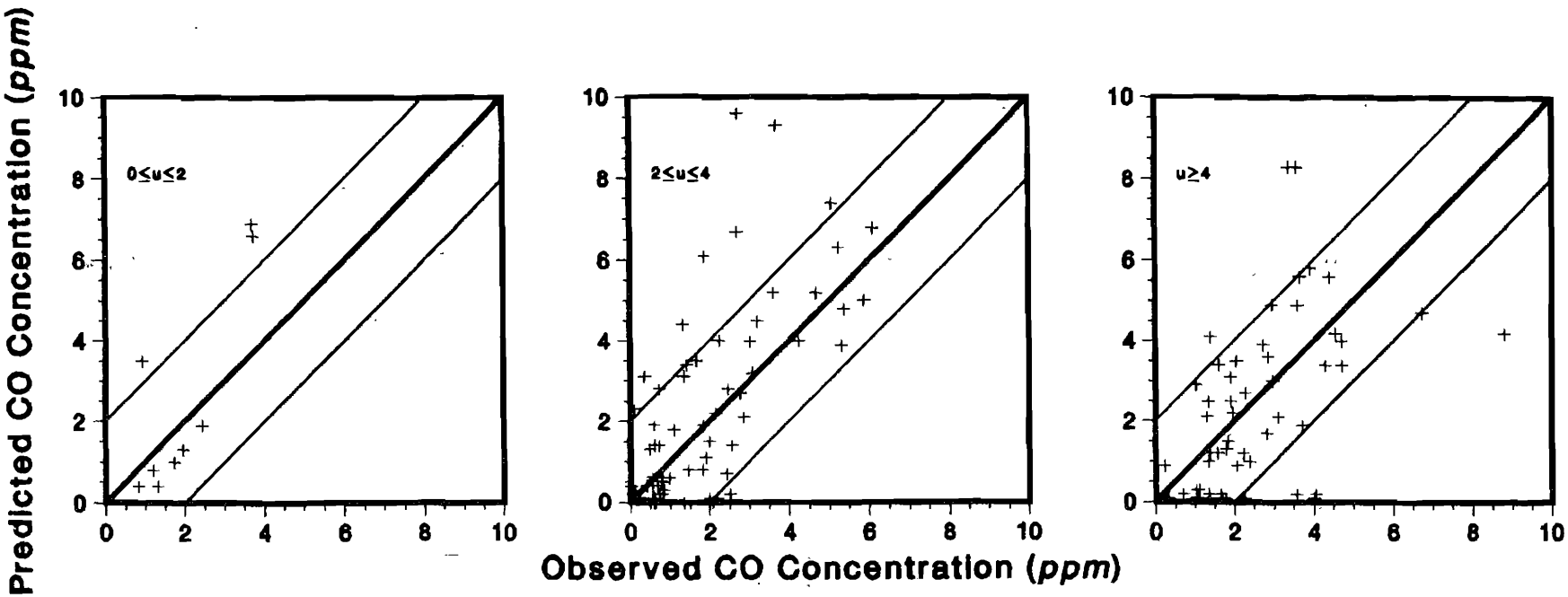
College Station Data Base



Scatterplots for Near Parallel Wind Cases

TEXIN2 Model CMA Operations & Design—MOBILE3

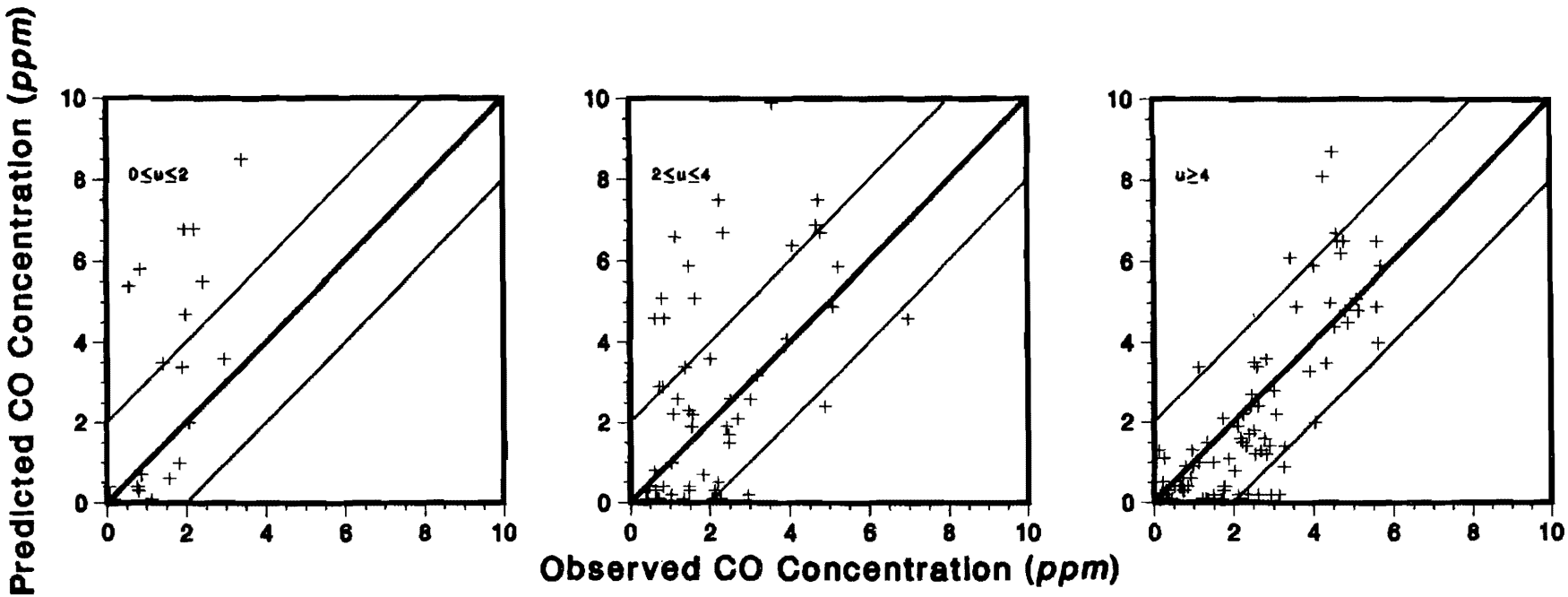
College Station Data Base



Scatterplots for Near 45° Wind Cases

TEXIN2 Model CMA Operations & Design—MOBILE3

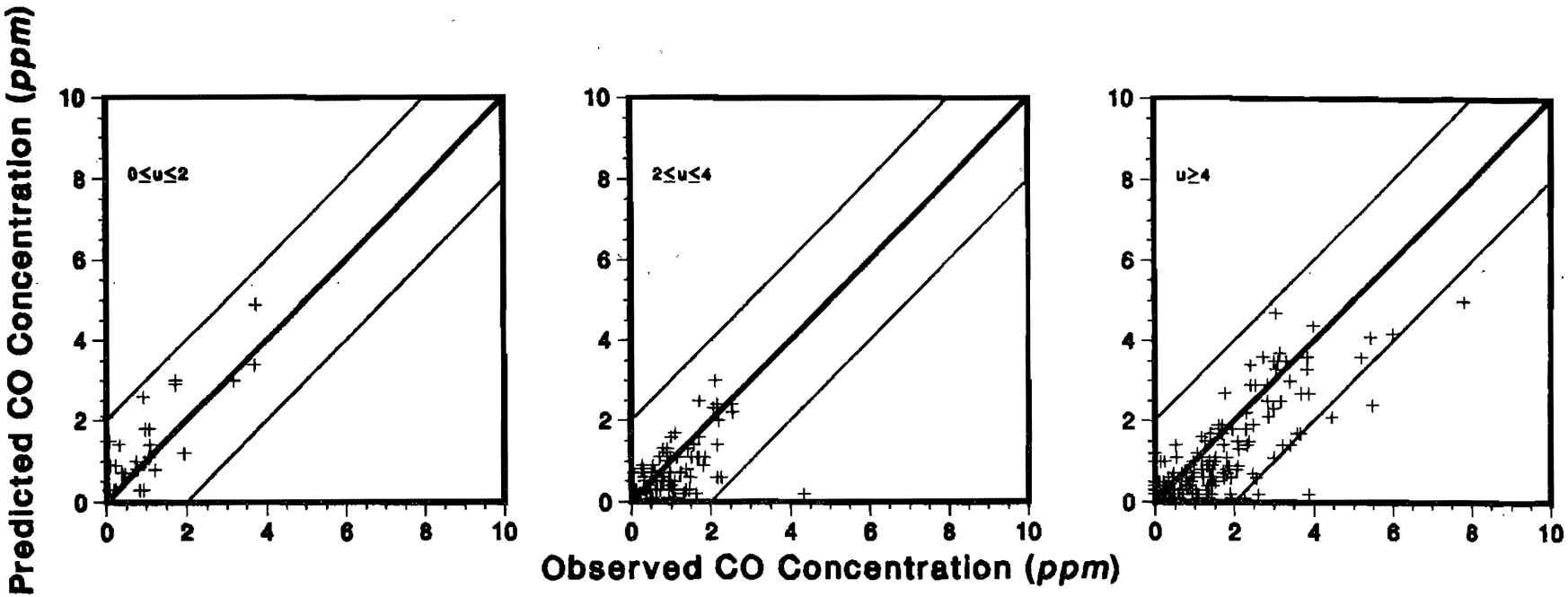
College Station Data Base



Scatterplots for Near Perpendicular Wind Cases

TEXIN2 Model CMA Operations & Design—MOBILE3

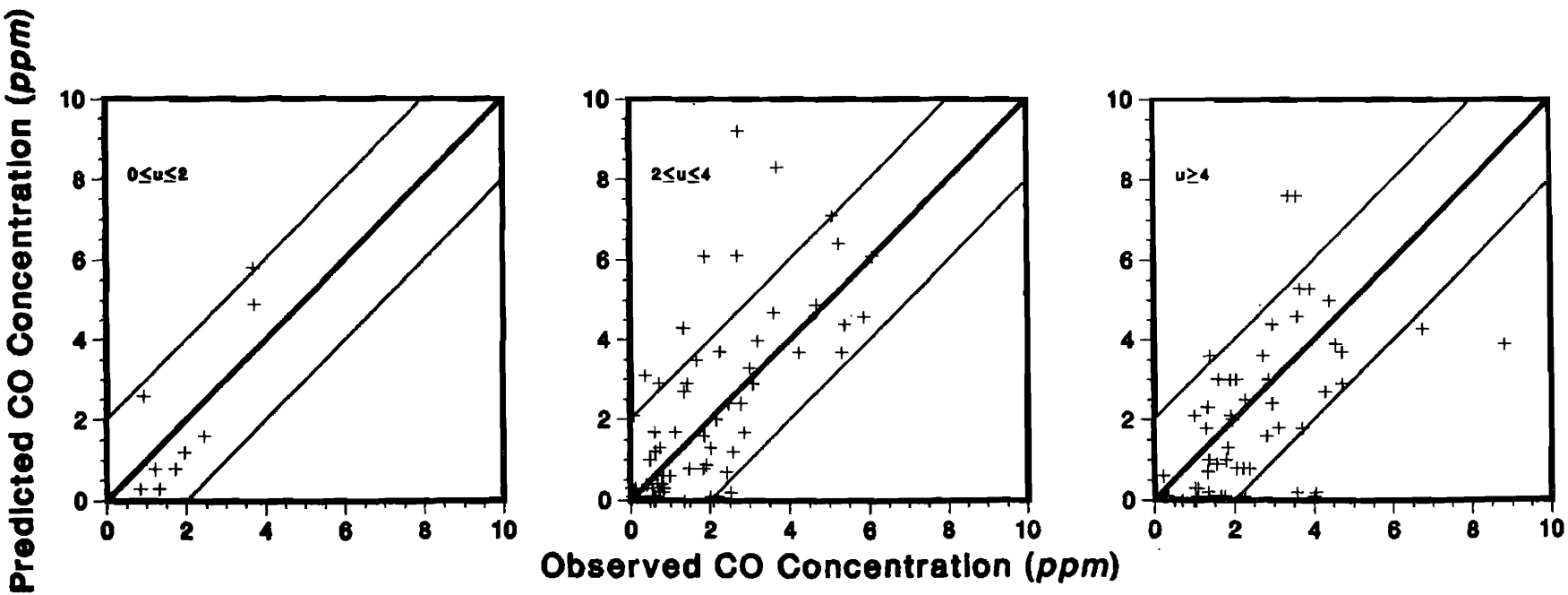
College Station Data Base



Scatterplots for Near Parallel Wind Cases

TEXIN2 Model CMA Planning—Short Cut Method

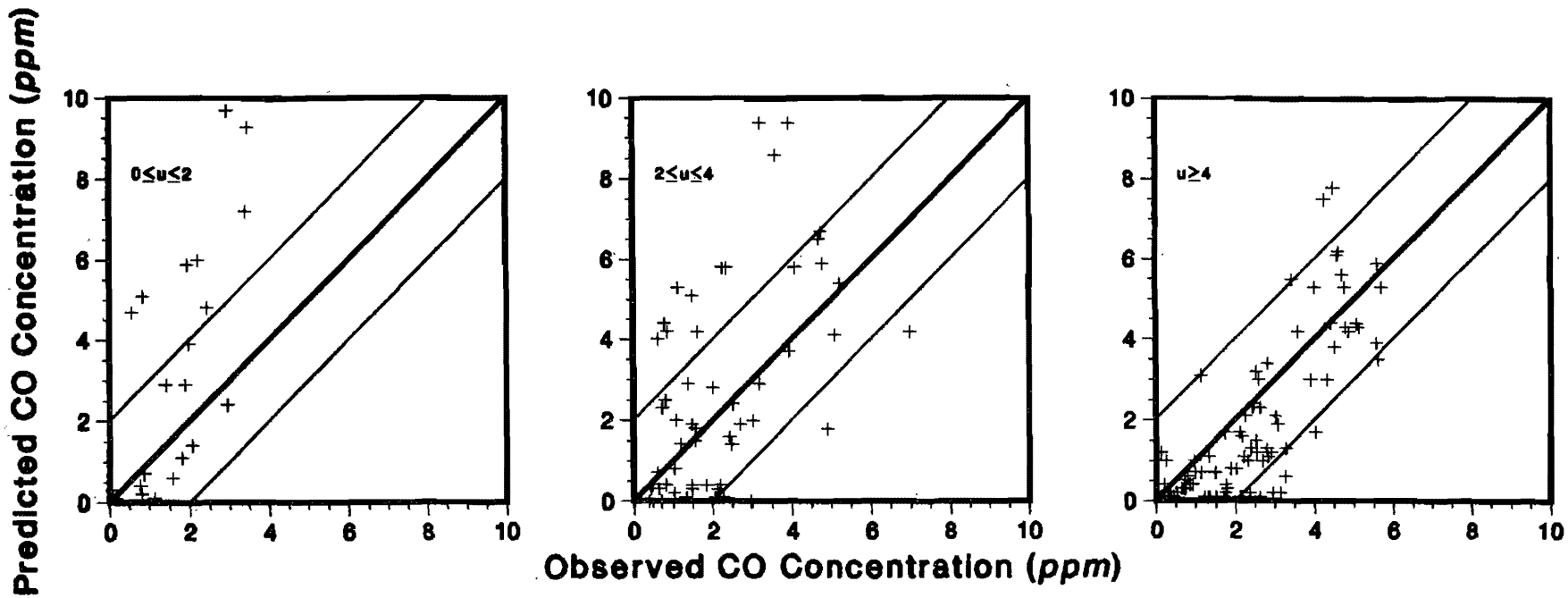
College Station Data Base



Scatterplots for Near 45° Wind Cases

TEXIN2 Model CMA Planning—Short Cut Method

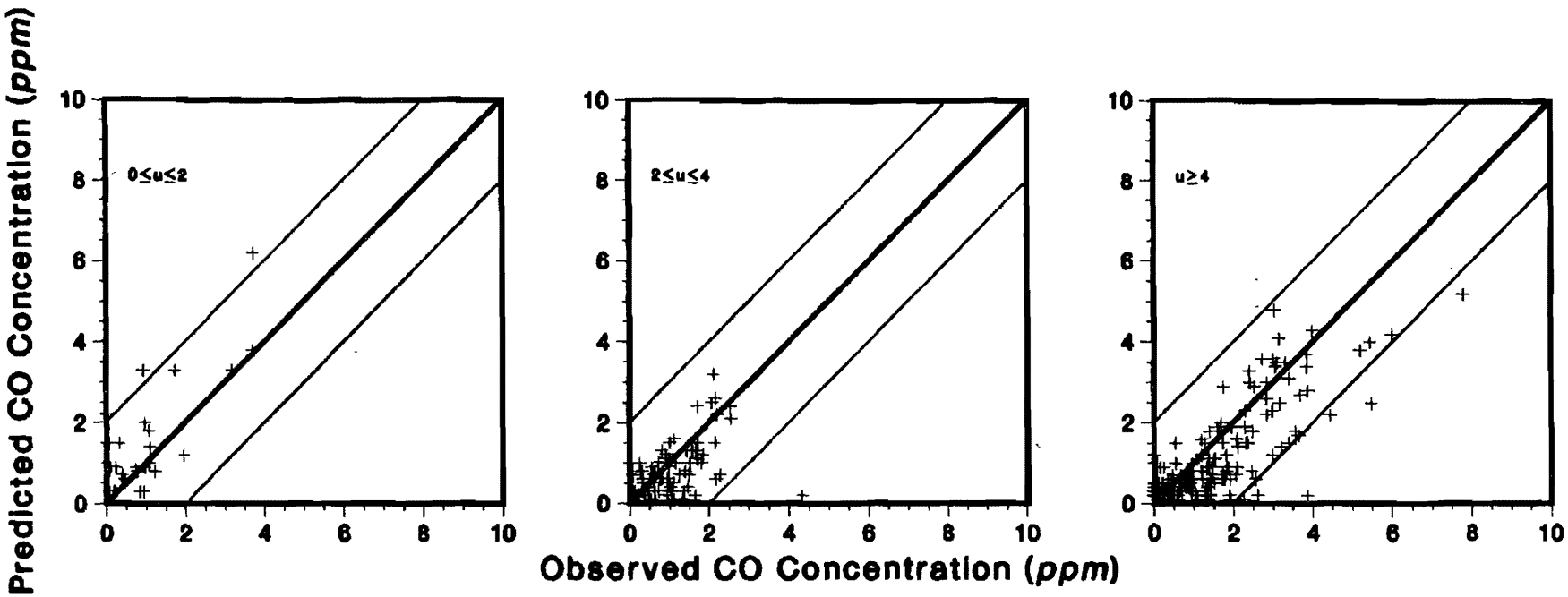
College Station Data Base



Scatterplots for Near Perpendicular Wind Cases

TEXIN2 Model CMA Planning—Short Cut Method

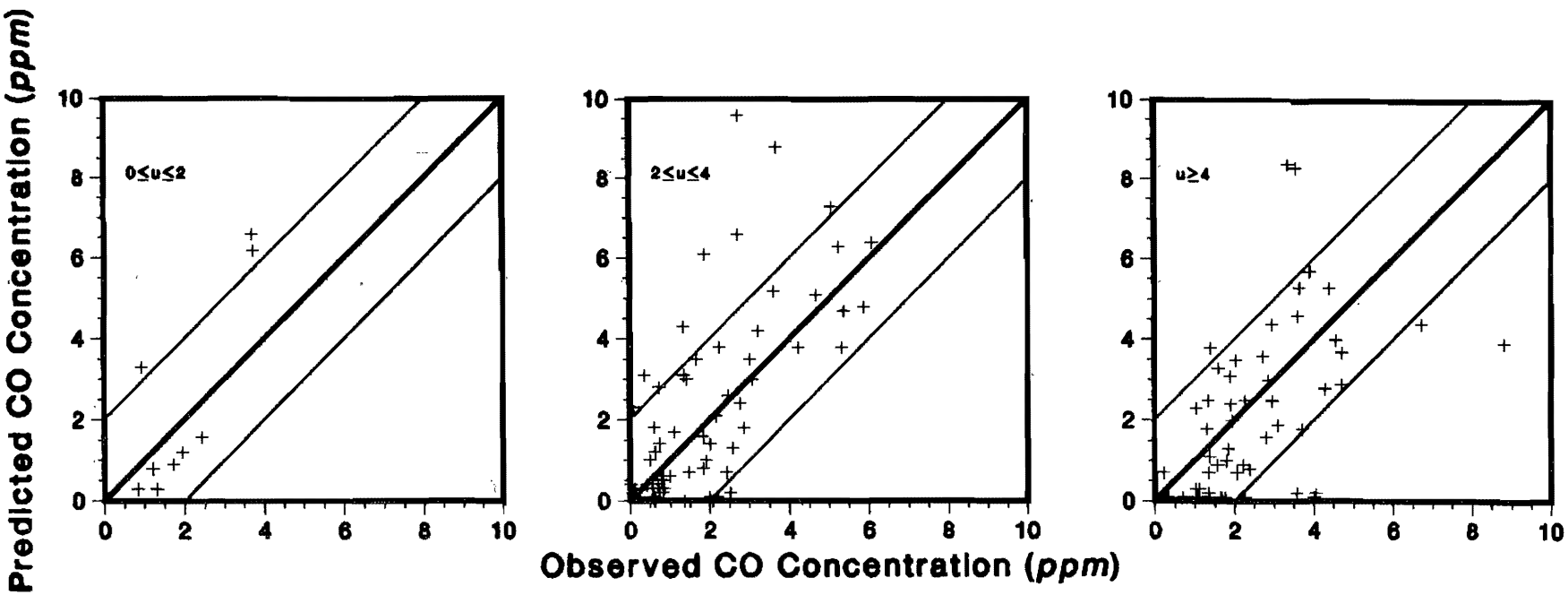
College Station Data Base



Scatterplots for Near Parallel Wind Cases

TEXIN2 Model CMA Planning—MOBILE3

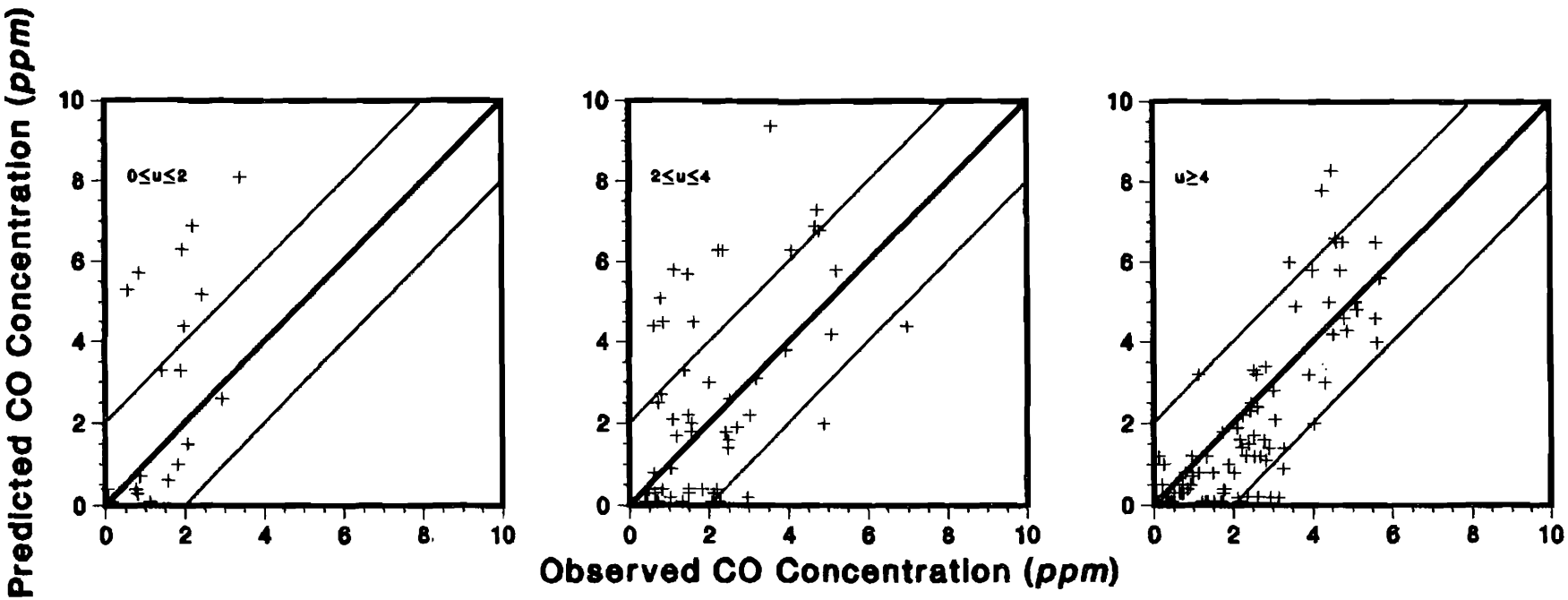
College Station Data Base



Scatterplots for Near 45° Wind Cases

TEXIN2 Model CMA Planning—MOBILE3

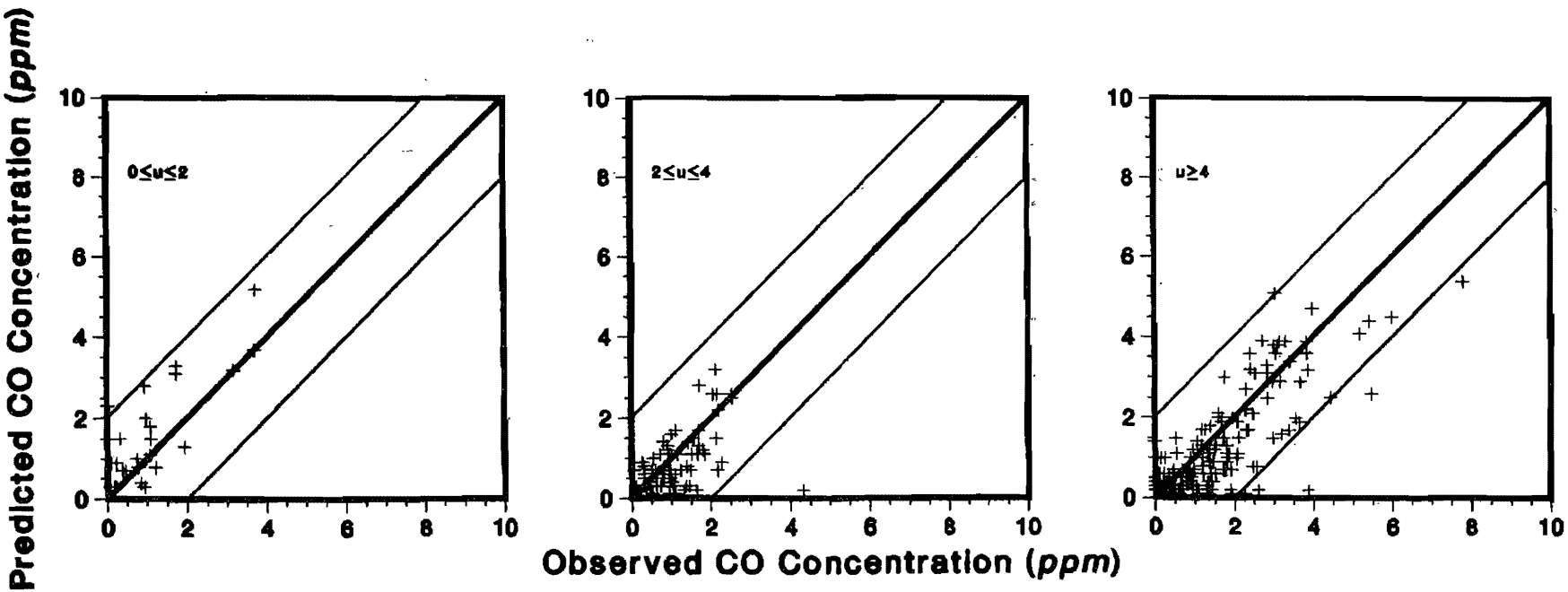
College Station Data Base



Scatterplots for Near Perpendicular Wind Cases

TEXIN2 Model CMA Planning—MOBILE3

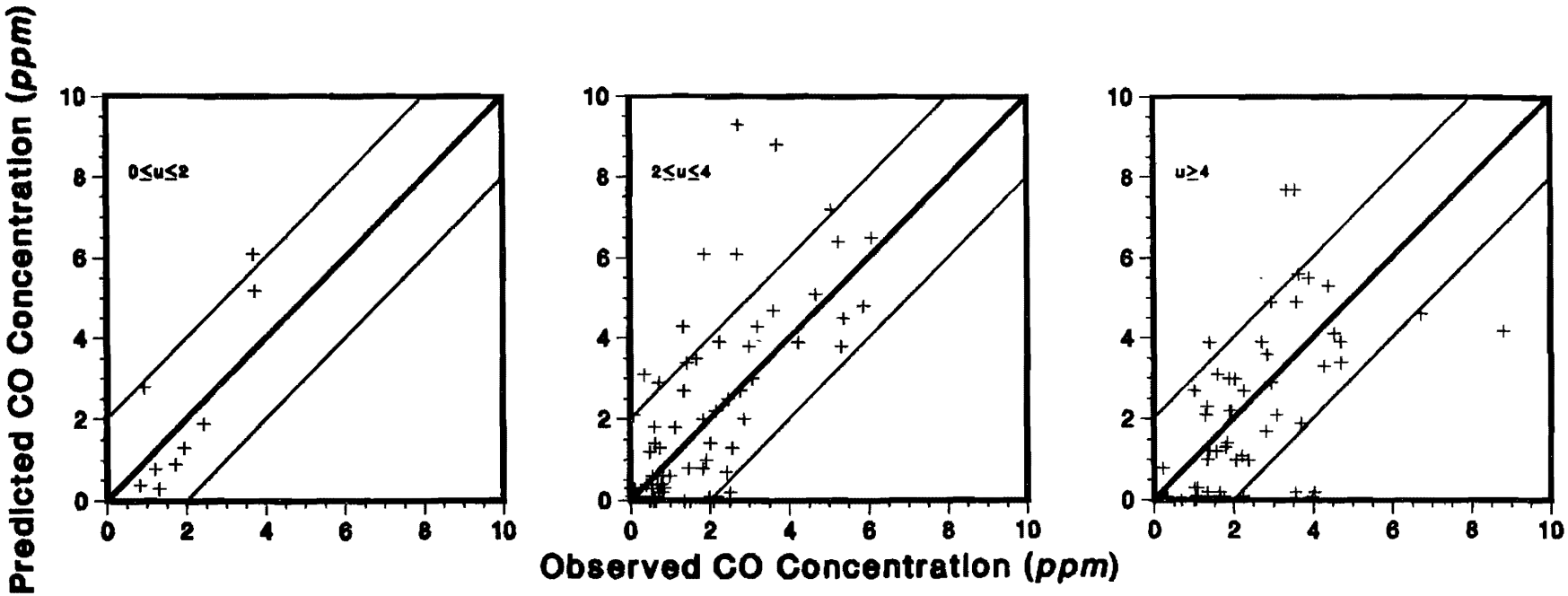
College Station Data Base



Scatterplots for Near Parallel Wind Cases

TEXIN2 Operations & Design—Short Cut Method

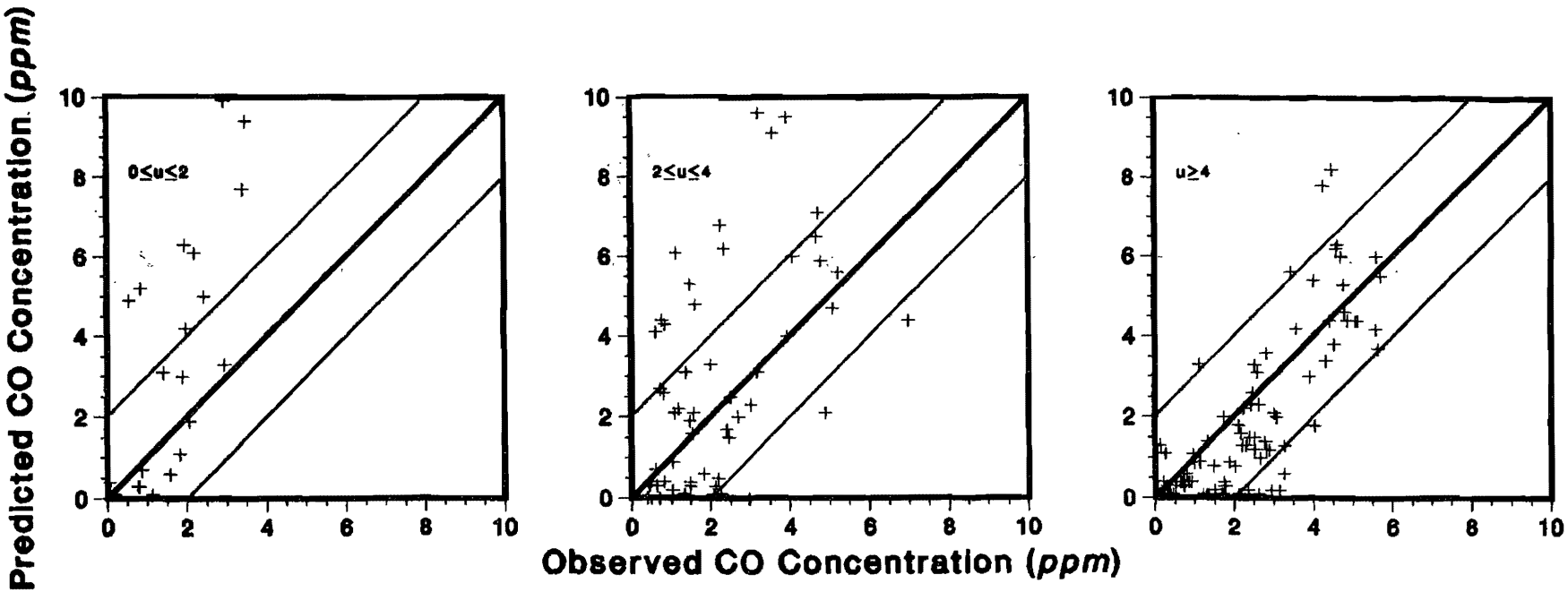
College Station Data Base



Scatterplots for Near 45° Wind Cases

TEXIN2 Operations & Design—Short Cut Method

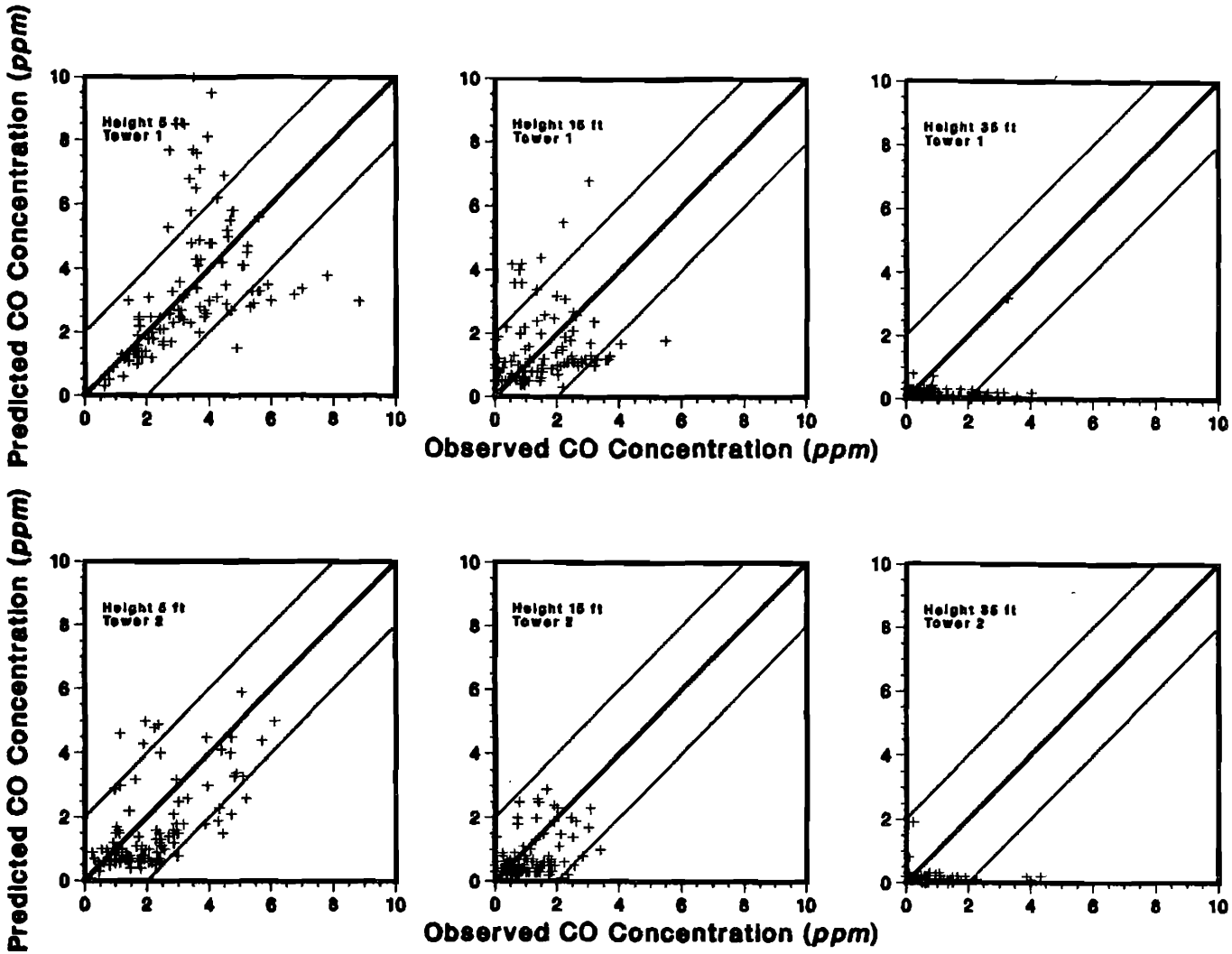
College Station Data Base



Scatterplots for Near Perpendicular Wind Cases

TEXIN2 Operations & Design—Short Cut Method

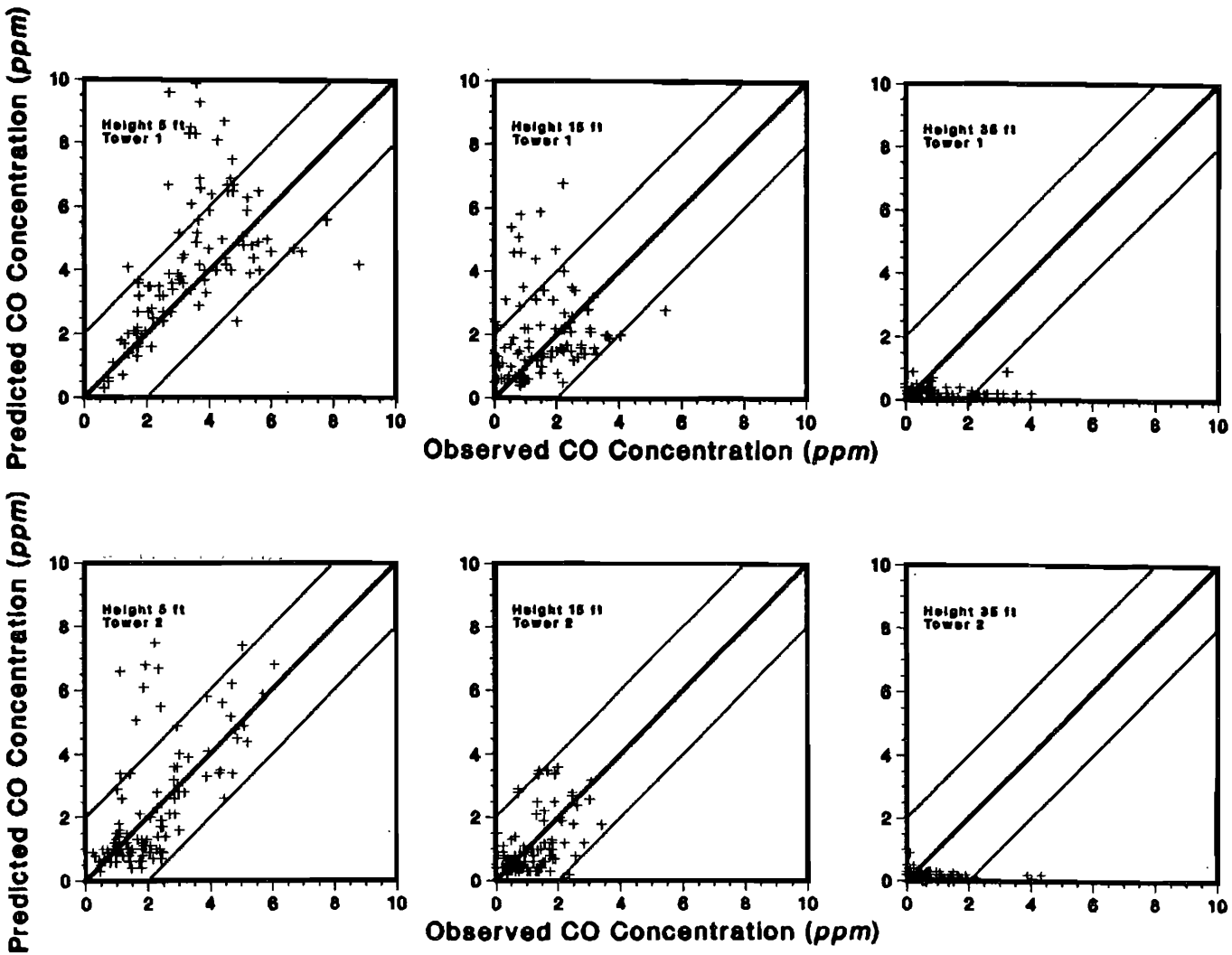
College Station Data Base



Scatterplots for Individual Receptors

Original TEXIN Model

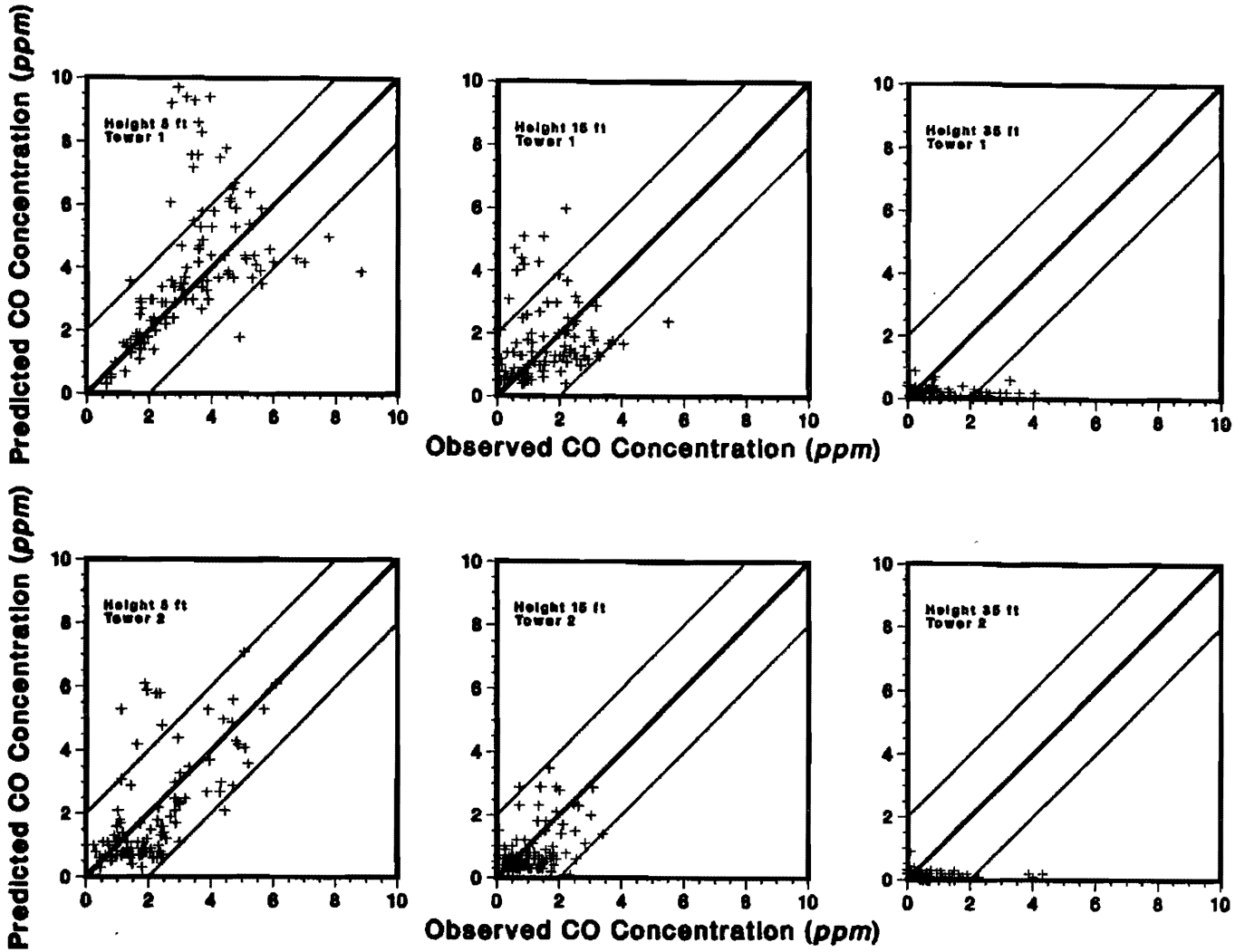
College Station Data Base



Scatterplots for Individual Receptors

TEXIN2 CMA Operations & Design—MOBILE3

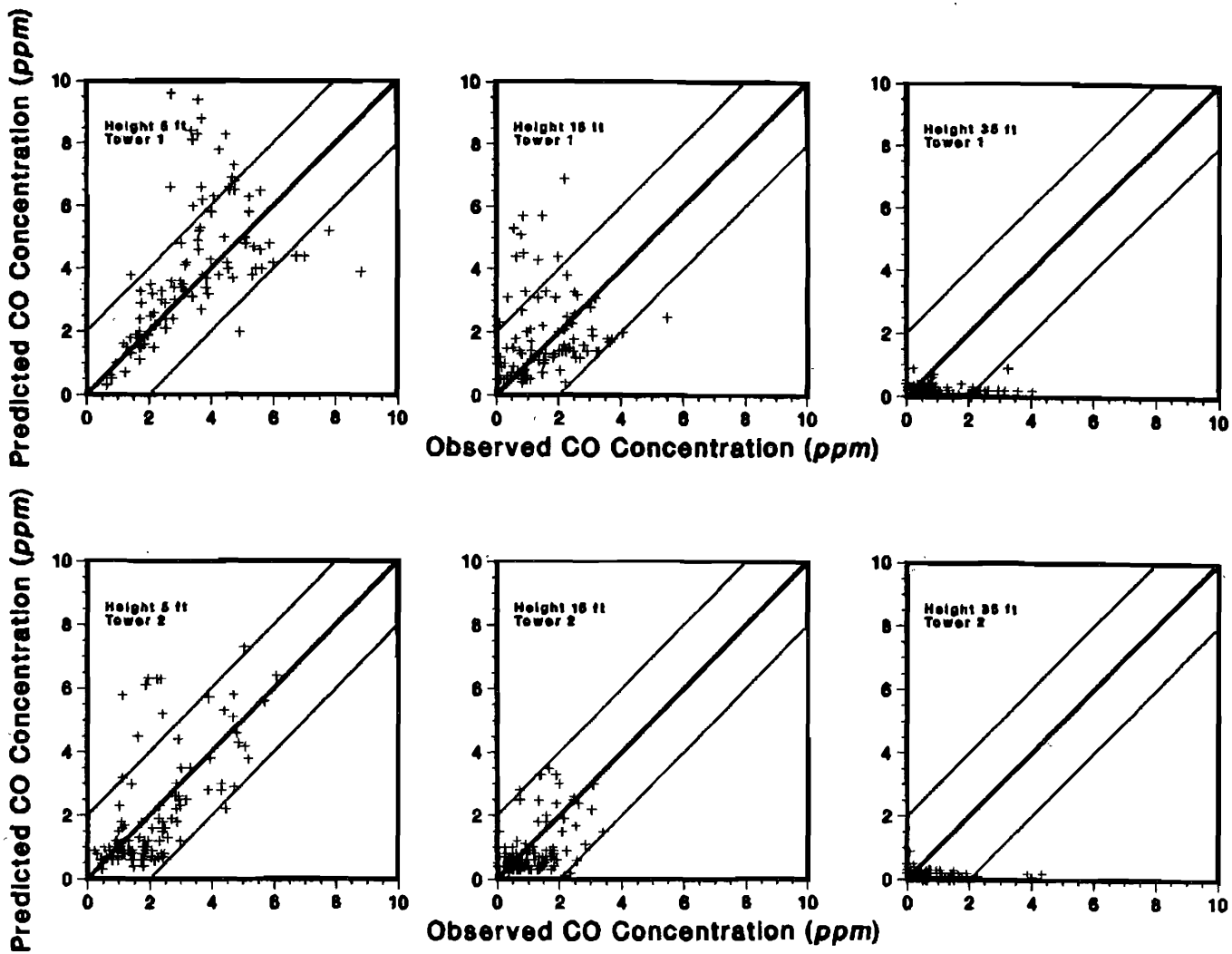
College Station Data Base



Scatterplots for Individual Receptors

TEXIN2 CMA Planning—Short Cut Method

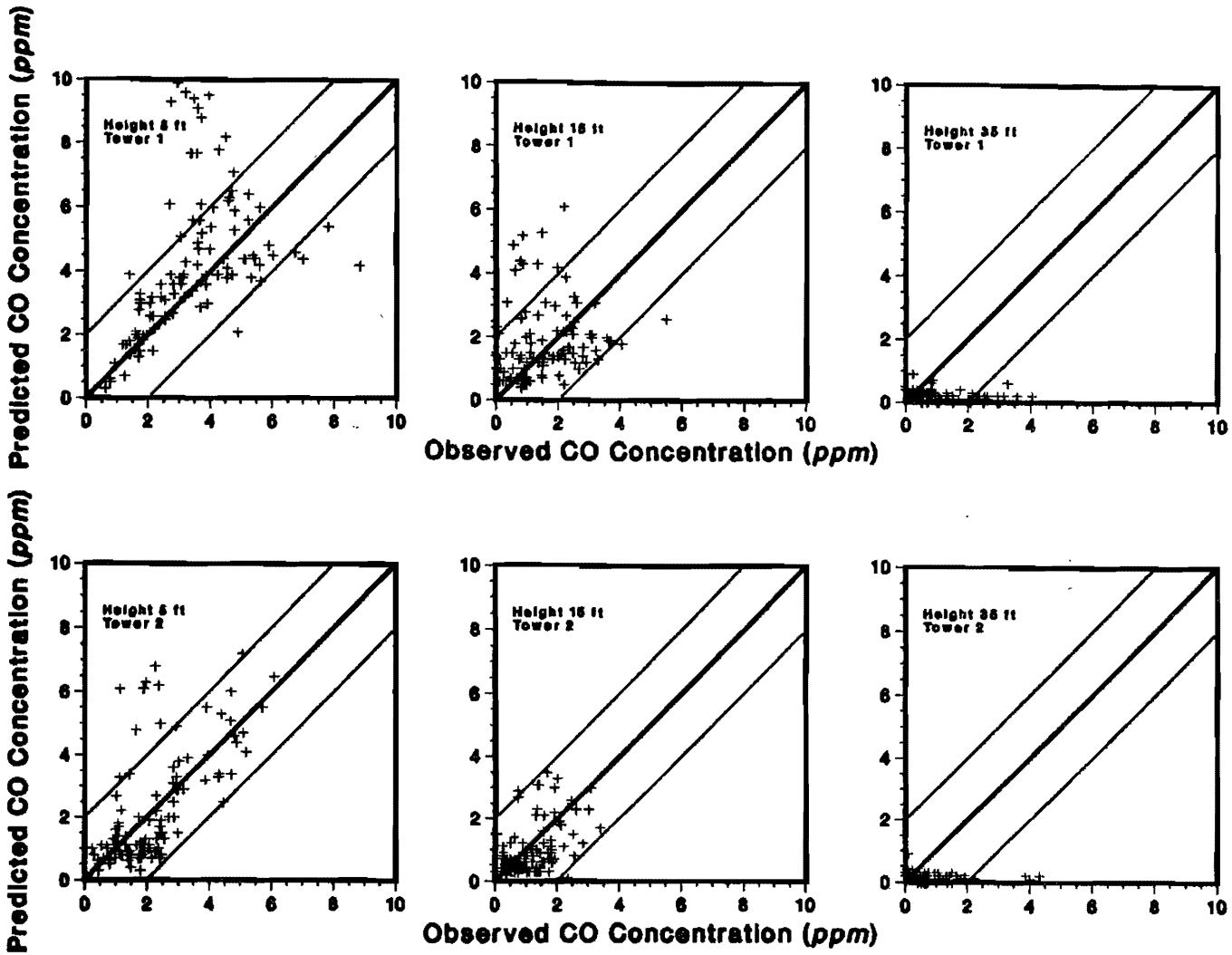
College Station Data Base



Scatterplots for Individual Receptors

TEXIN2 CMA Planning—MOBILE3

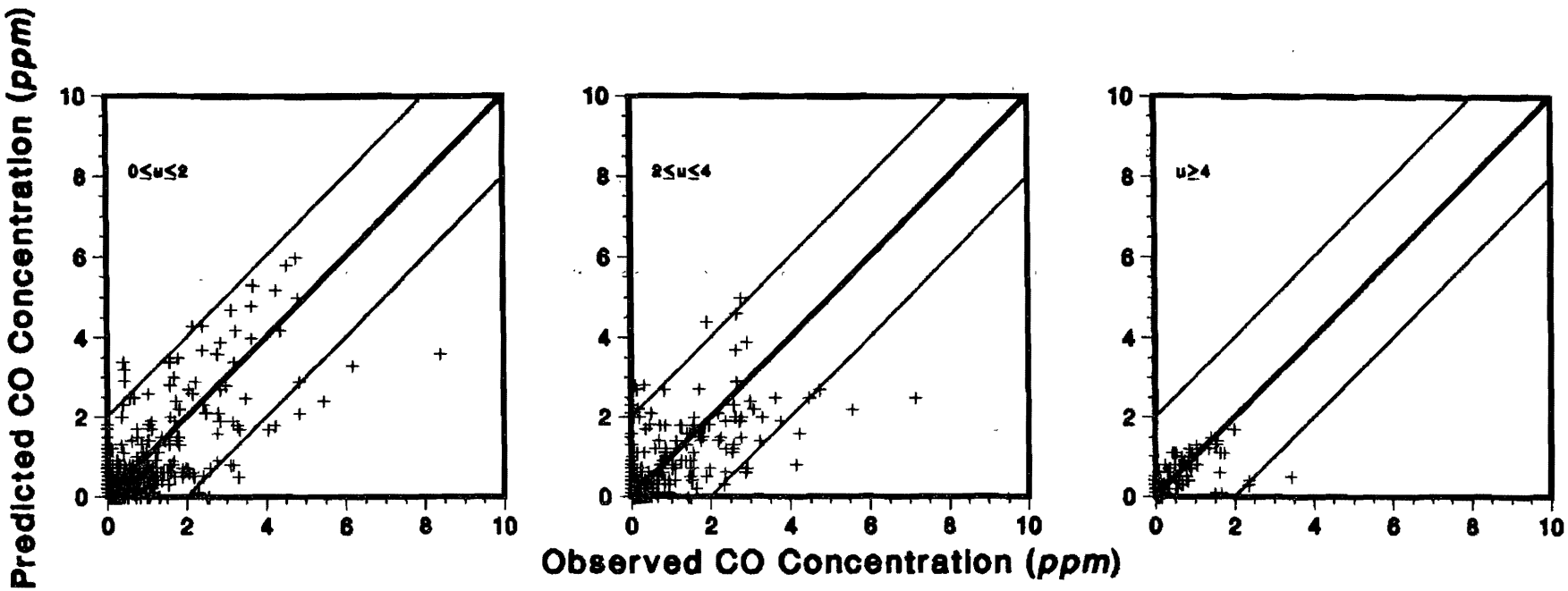
College Station Data Base



Scatterplots for Individual Receptors

TEXIN2 CMA Operations & Design—Short Cut Method

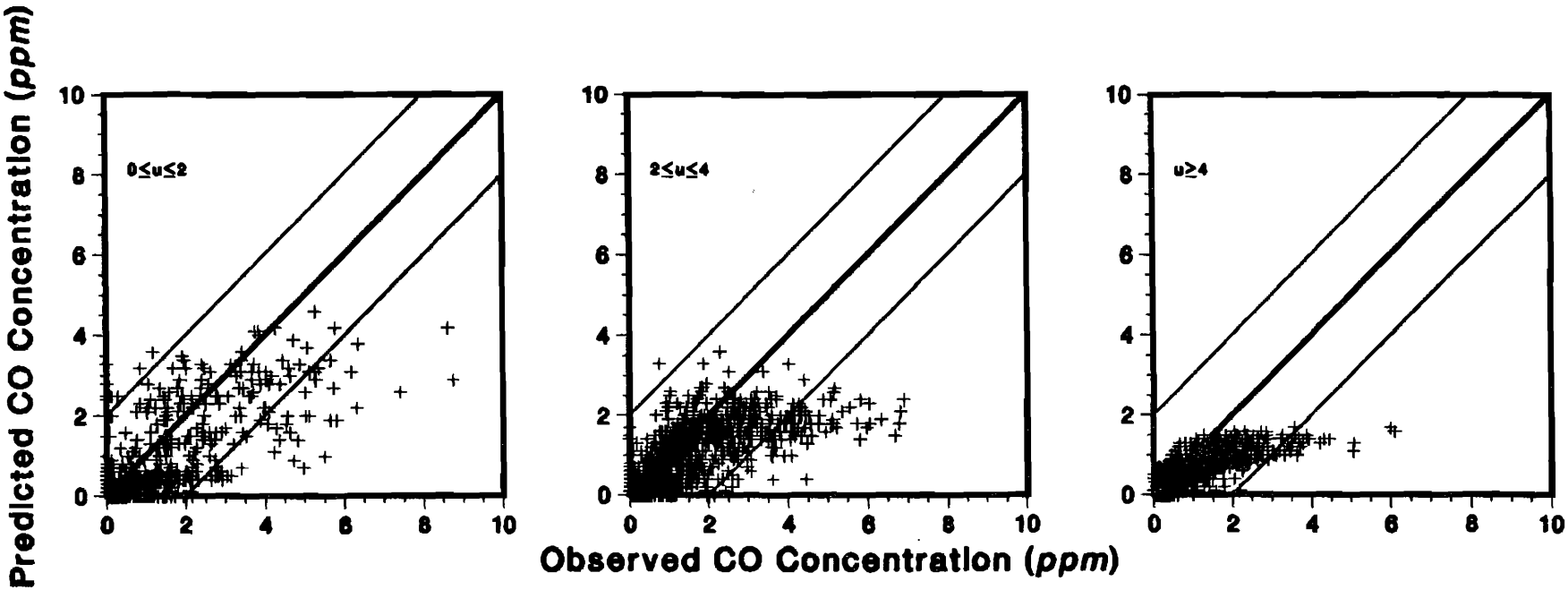
College Station Data Base



Scatterplots for Near Parallel Wind Cases

Original TEXIN Model

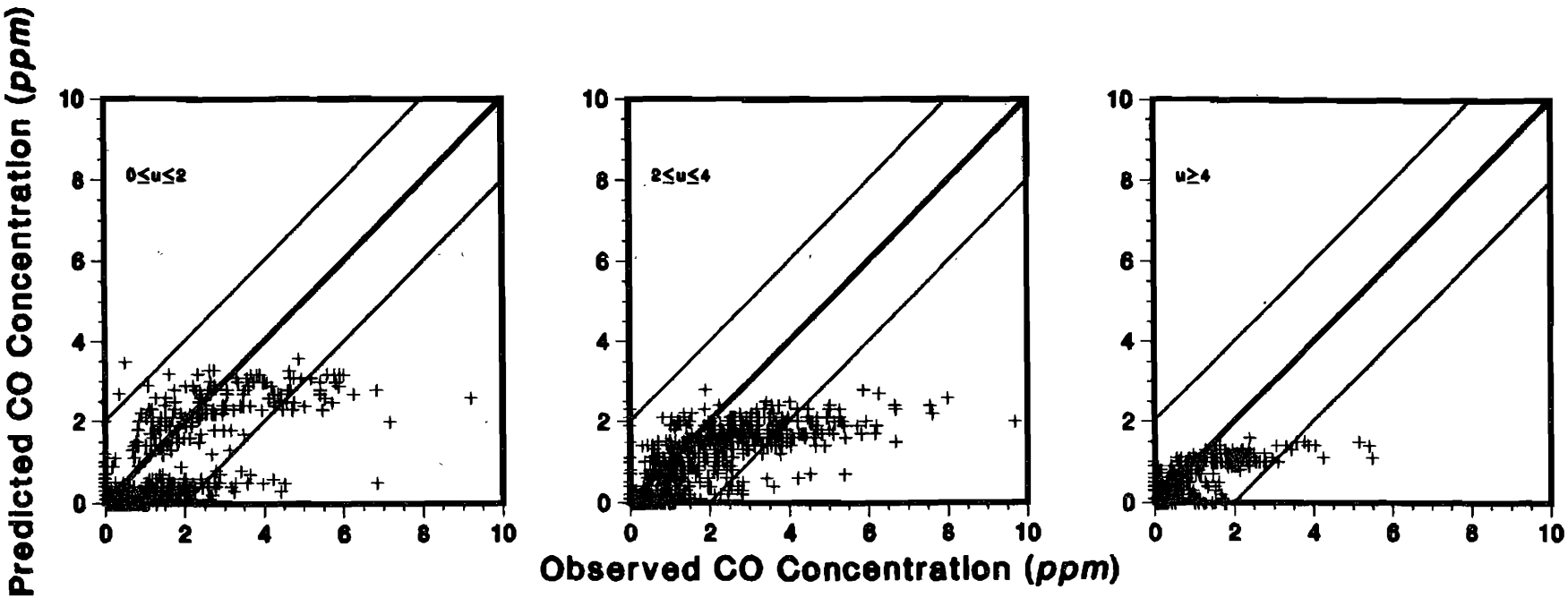
California Data Base



Scatterplots for Near 45° Wind Cases

Original TEXIN Model

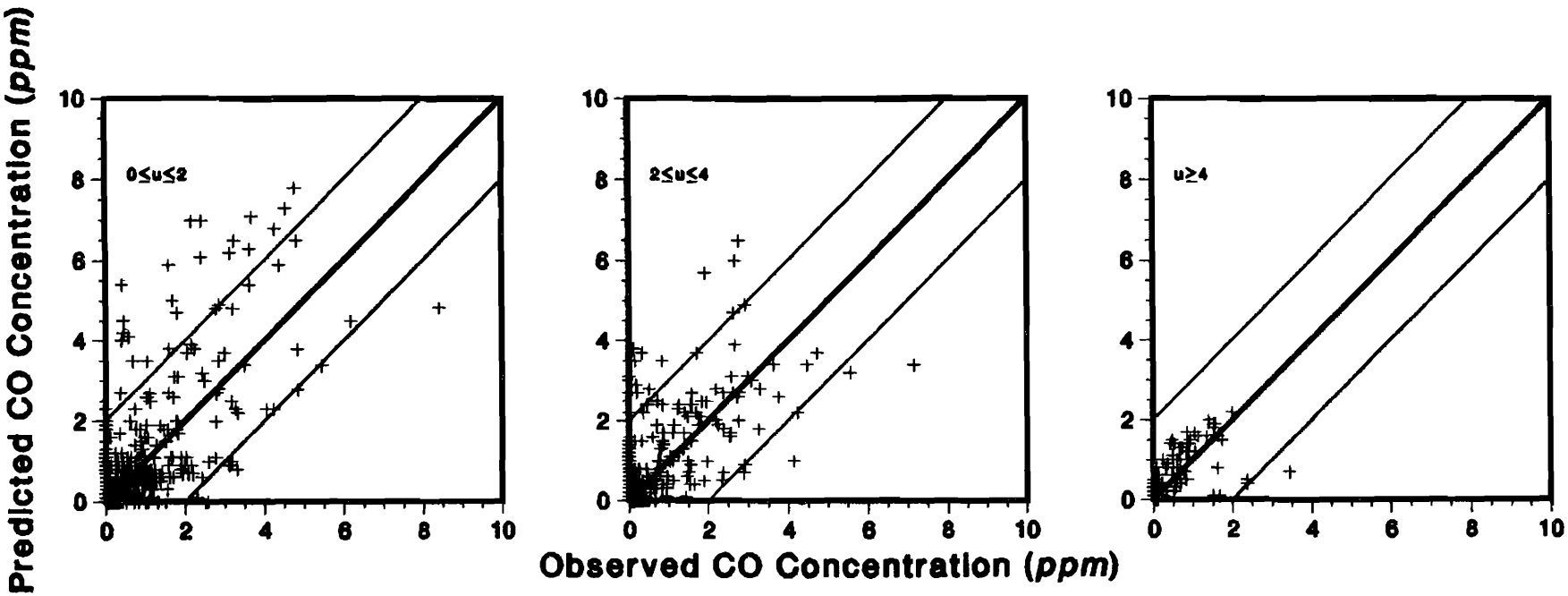
California Data Base



Scatterplots for Near Perpendicular Wind Cases

Original TEXIN Model

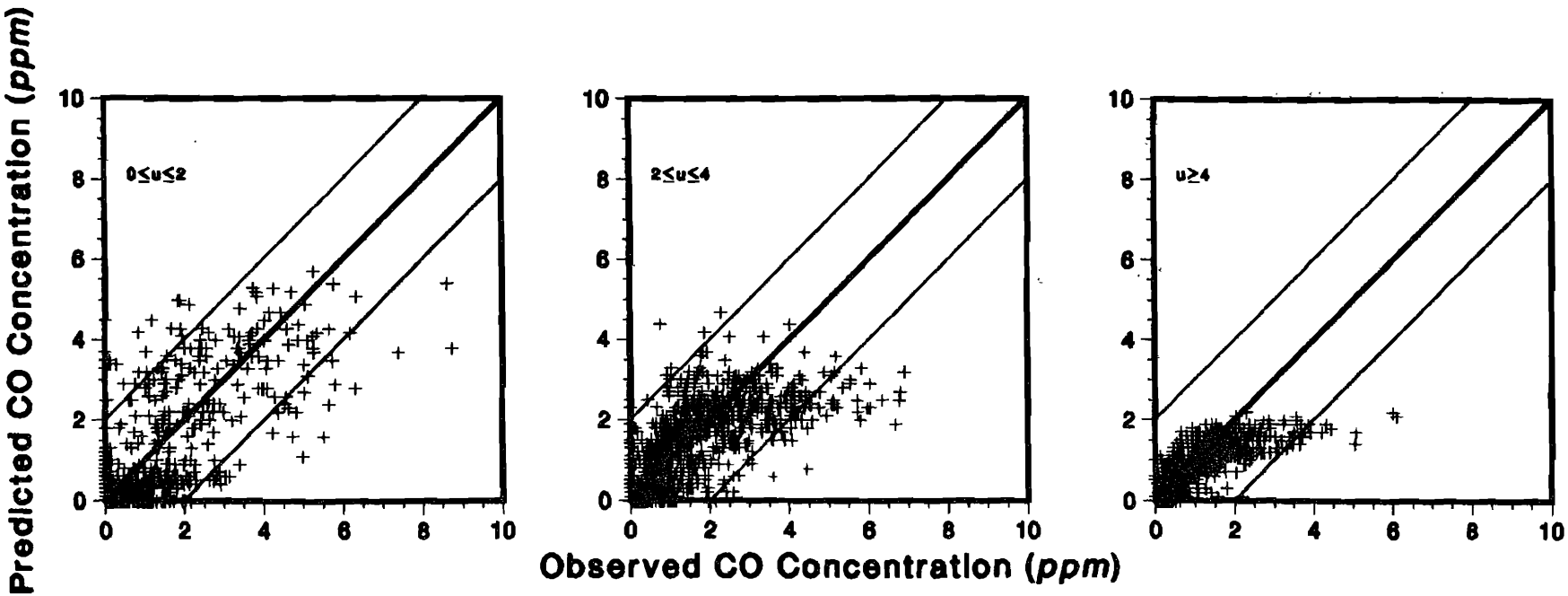
California Data Base



Scatterplots for Near Parallel Wind Cases

TEXIN2 Model CMA Operations & Design—MOBILE3

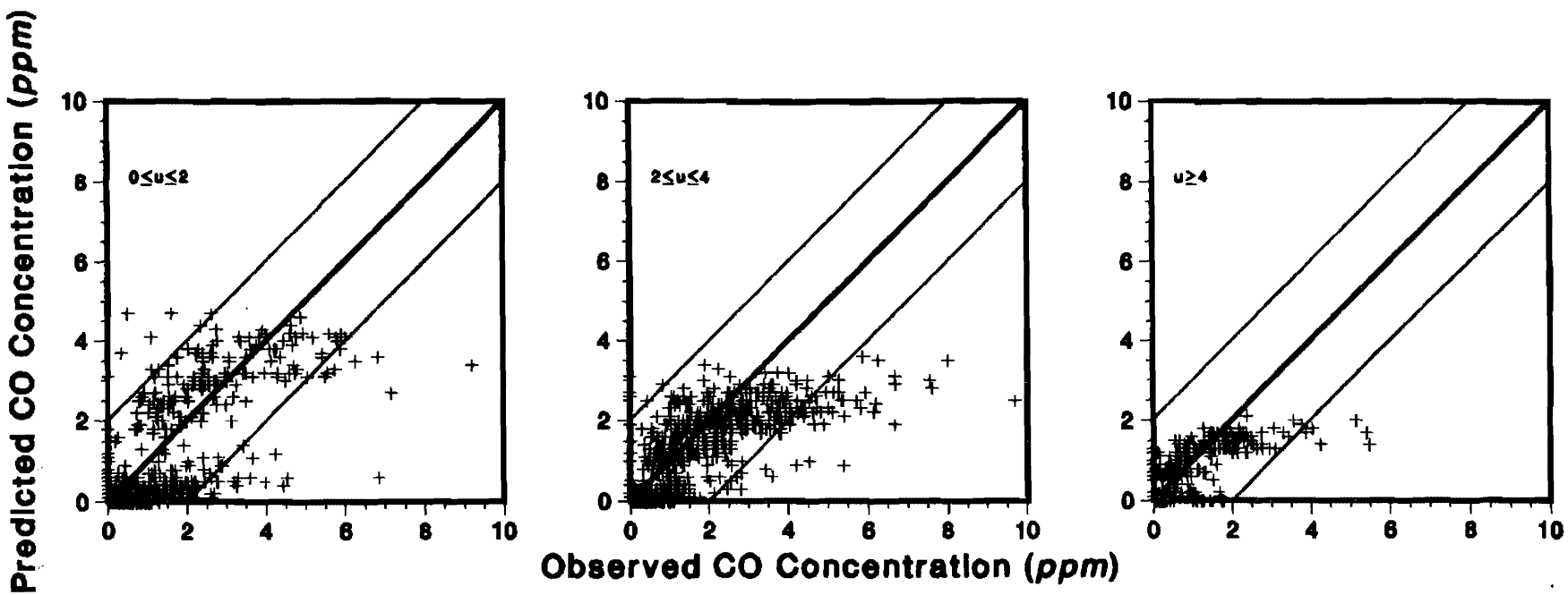
California Data Base



Scatterplots for Near 45° Wind Cases

TEXIN2 Model CMA Operations & Design—MOBILE3

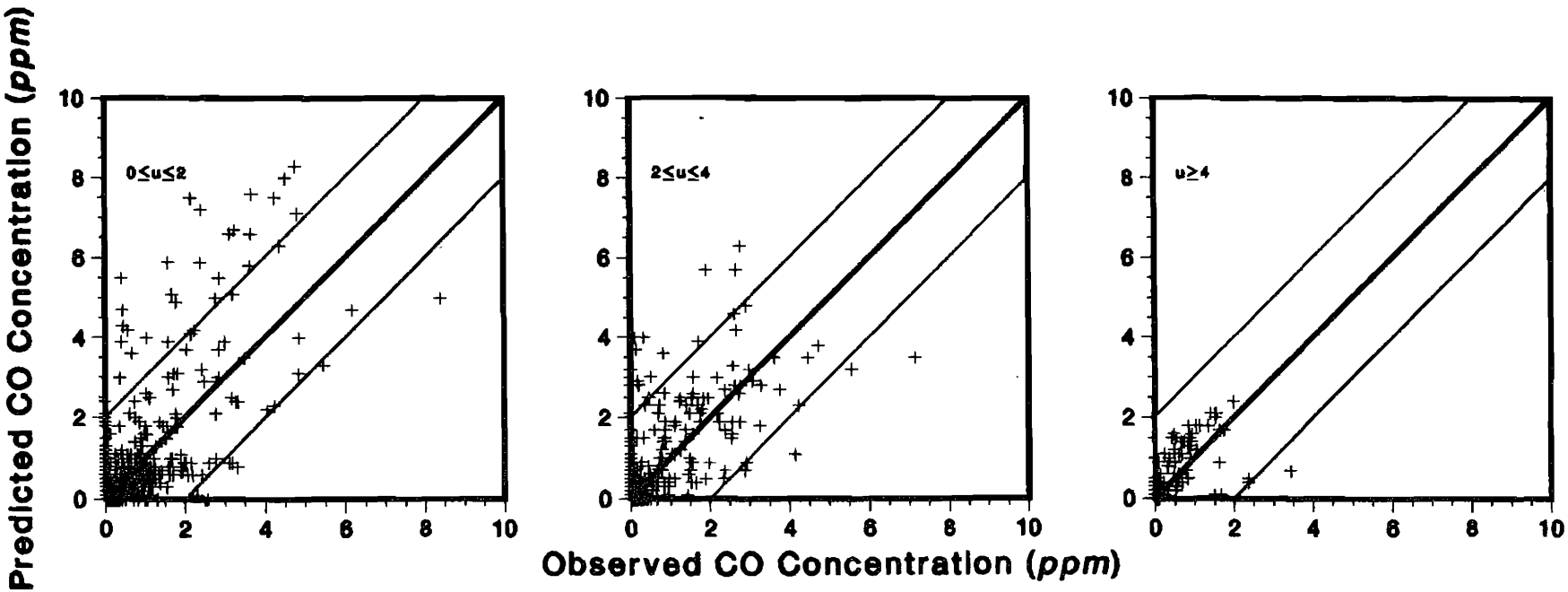
California Data Base



Scatterplots for Near Perpendicular Wind Cases

TEXIN2 Model CMA Operations & Design—MOBILE3

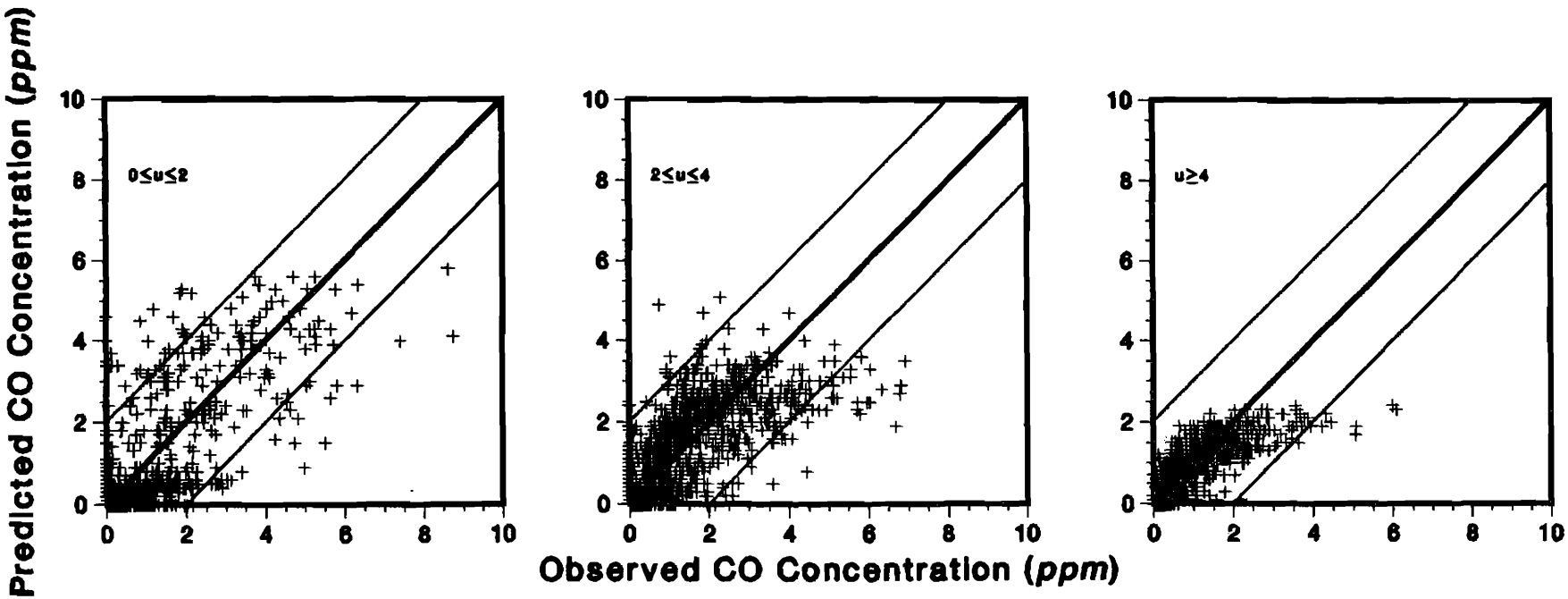
California Data Base



Scatterplots for Near Parallel Wind Cases

TEXIN2 Model CMA Planning—Short Cut Method

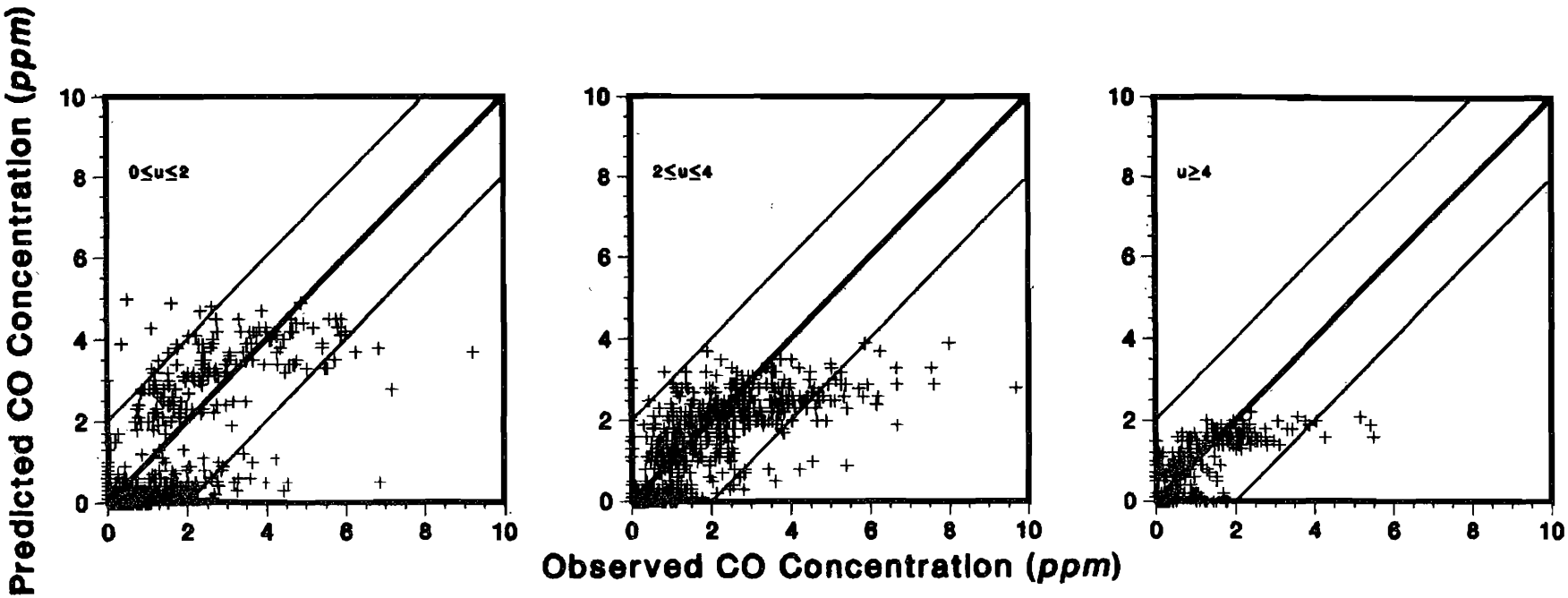
California Data Base



Scatterplots for Near 45° Wind Cases

TEXIN2 Model CMA Planning—Short Cut Method

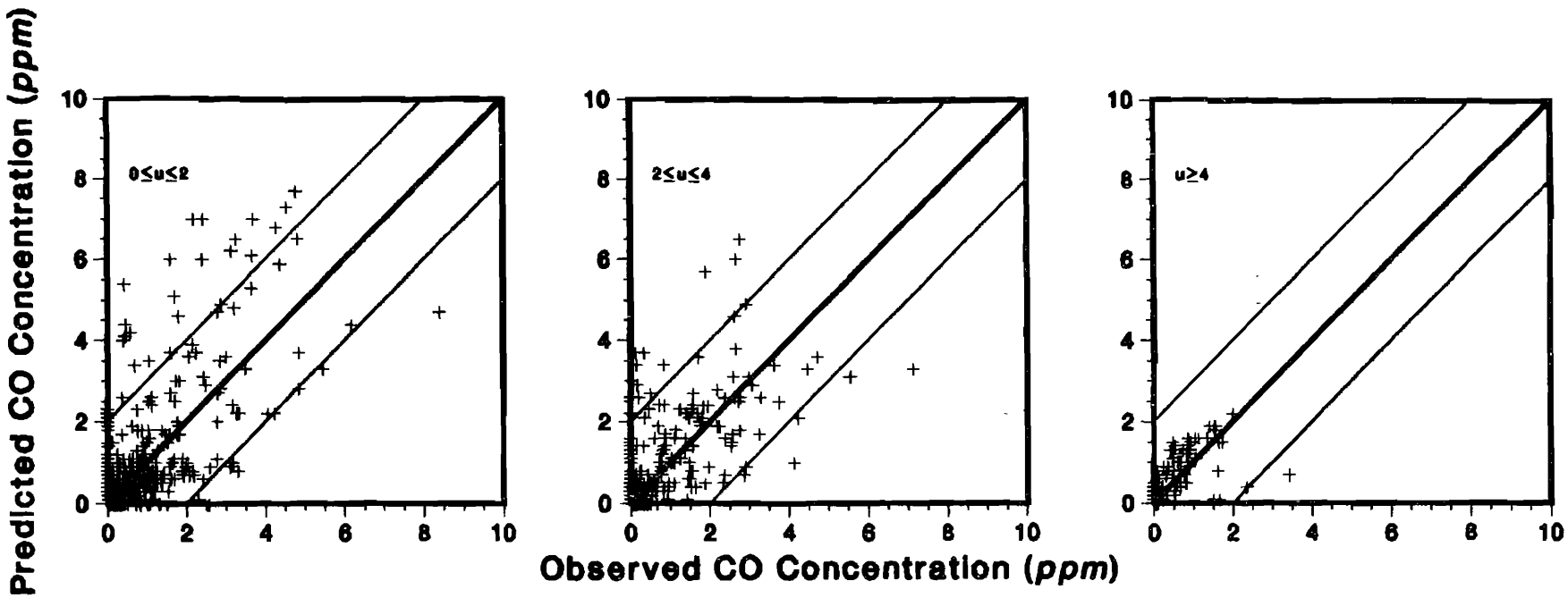
California Data Base



Scatterplots for Near Perpendicular Wind Cases

TEXIN2 Model CMA Planning—Short Cut Method

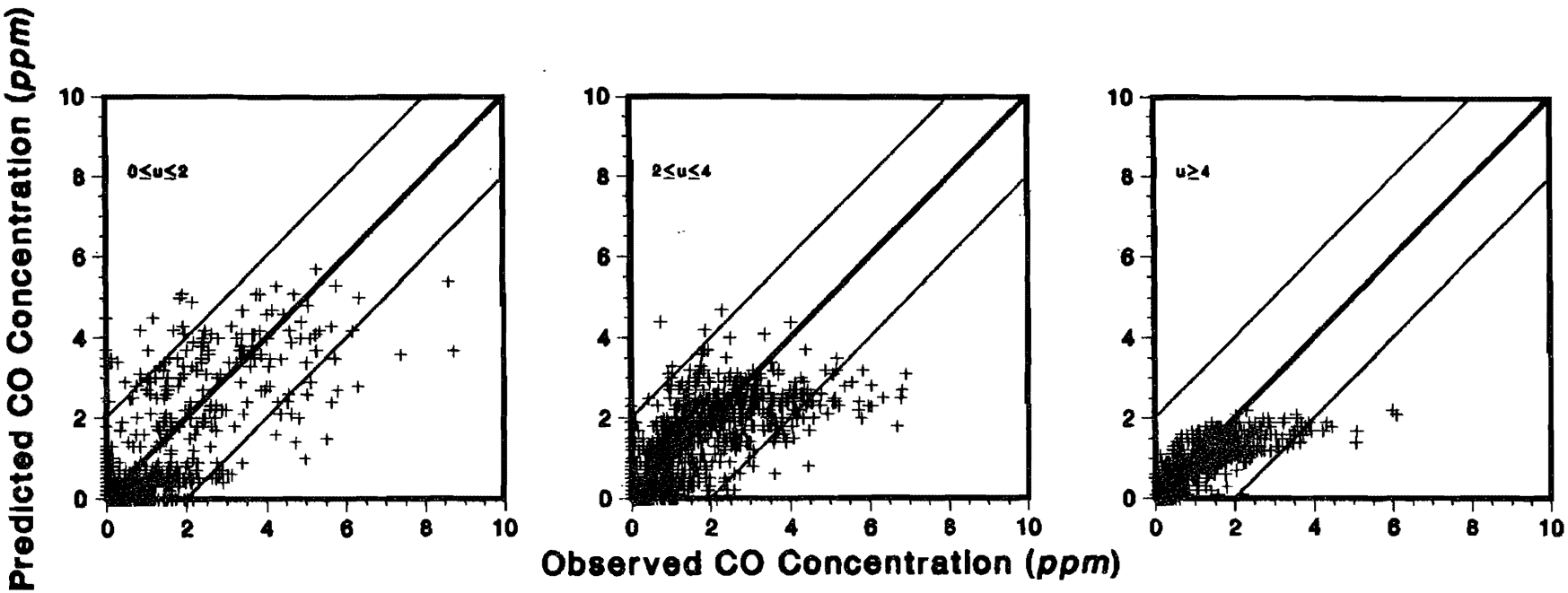
California Data Base



Scatterplots for Near Parallel Wind Cases

TEXIN2 Model CMA Planning—MOBILE3

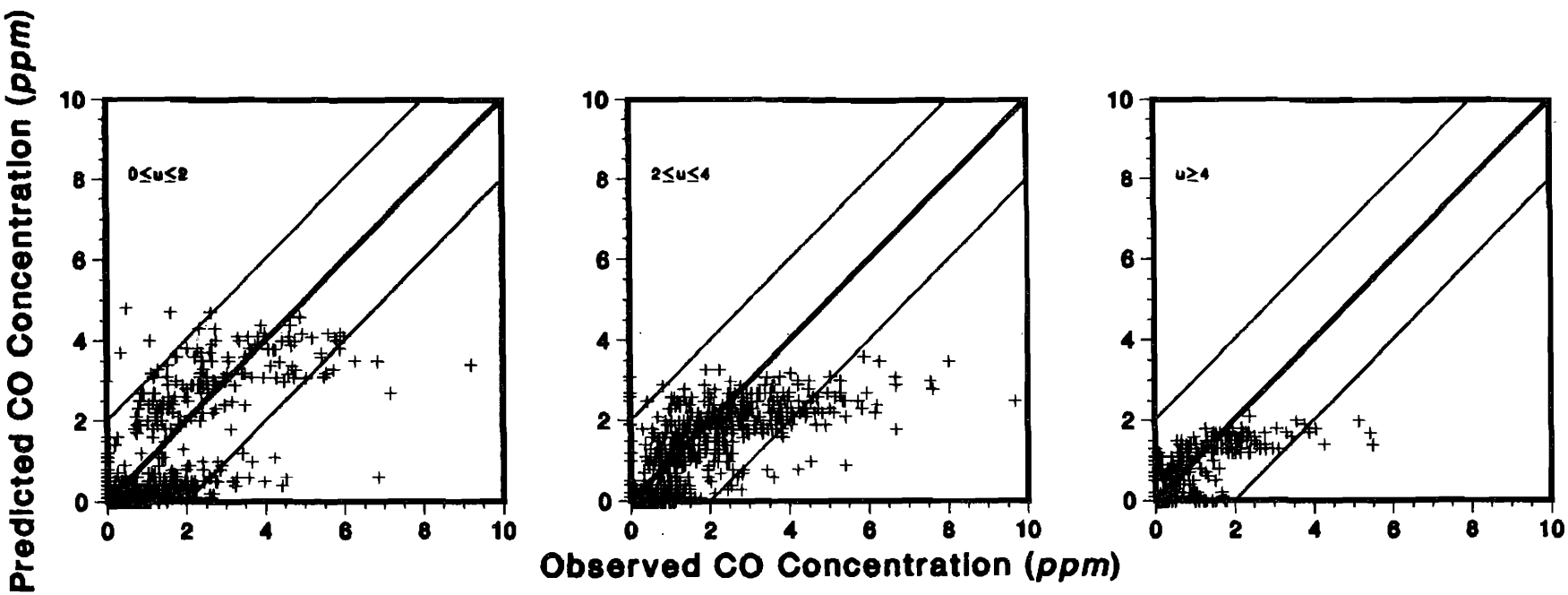
California Data Base



Scatterplots for Near 45° Wind Cases

TEXIN2 Model CMA Planning—MOBILE3

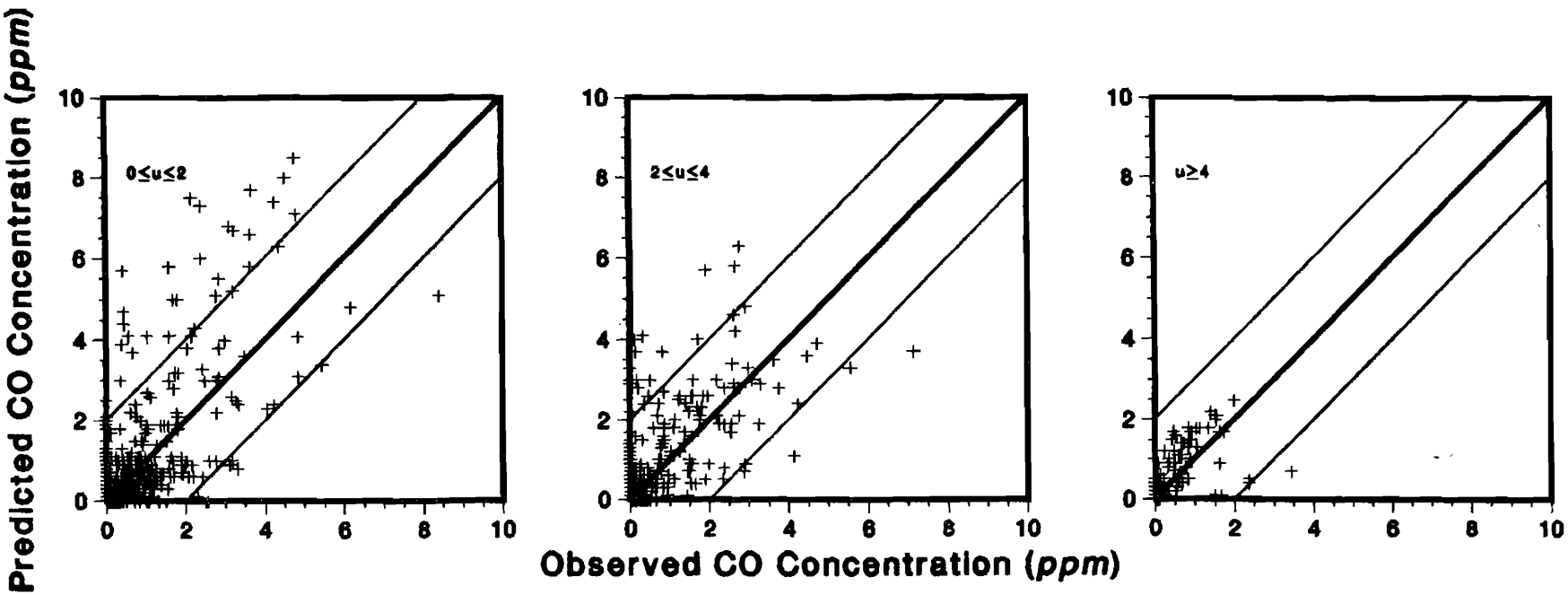
California Data Base



Scatterplots for Near Perpendicular Wind Cases

TEXIN2 Model CMA Planning—MOBILE3

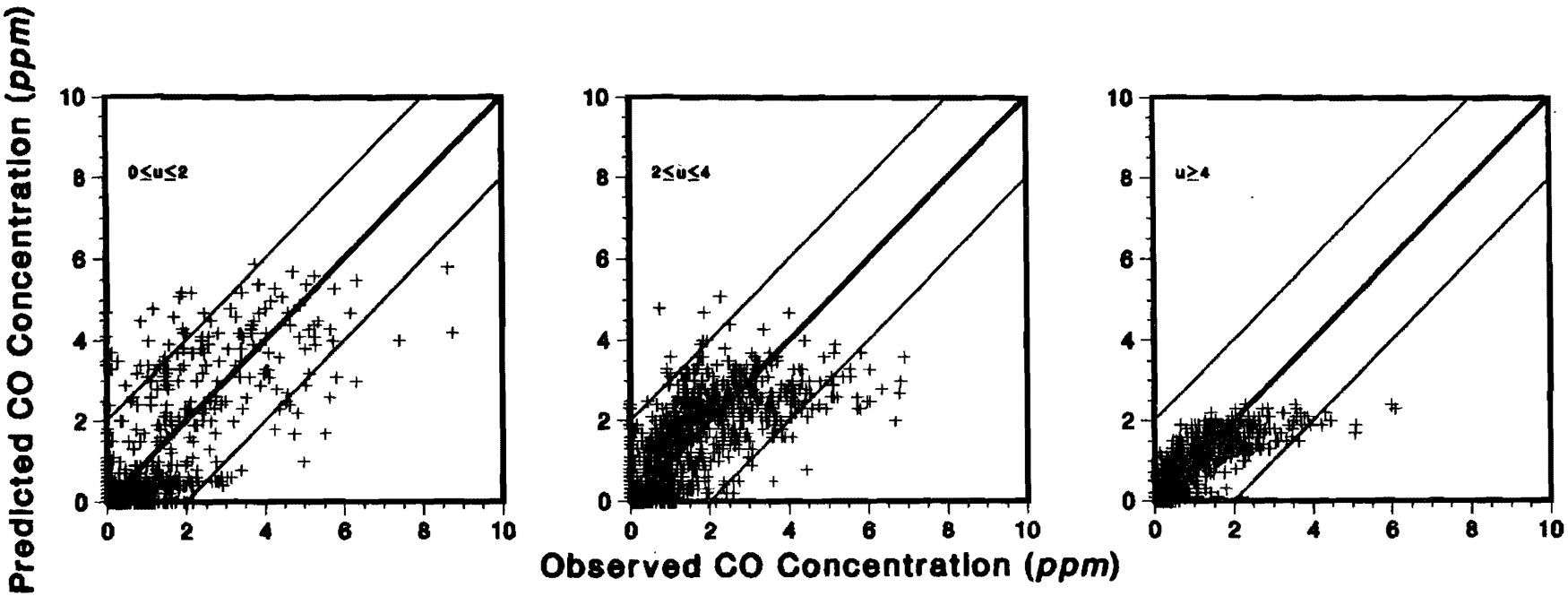
California Data Base



Scatterplots for Near Parallel Wind Cases

TEXIN2 Operations & Design—Short Cut Method

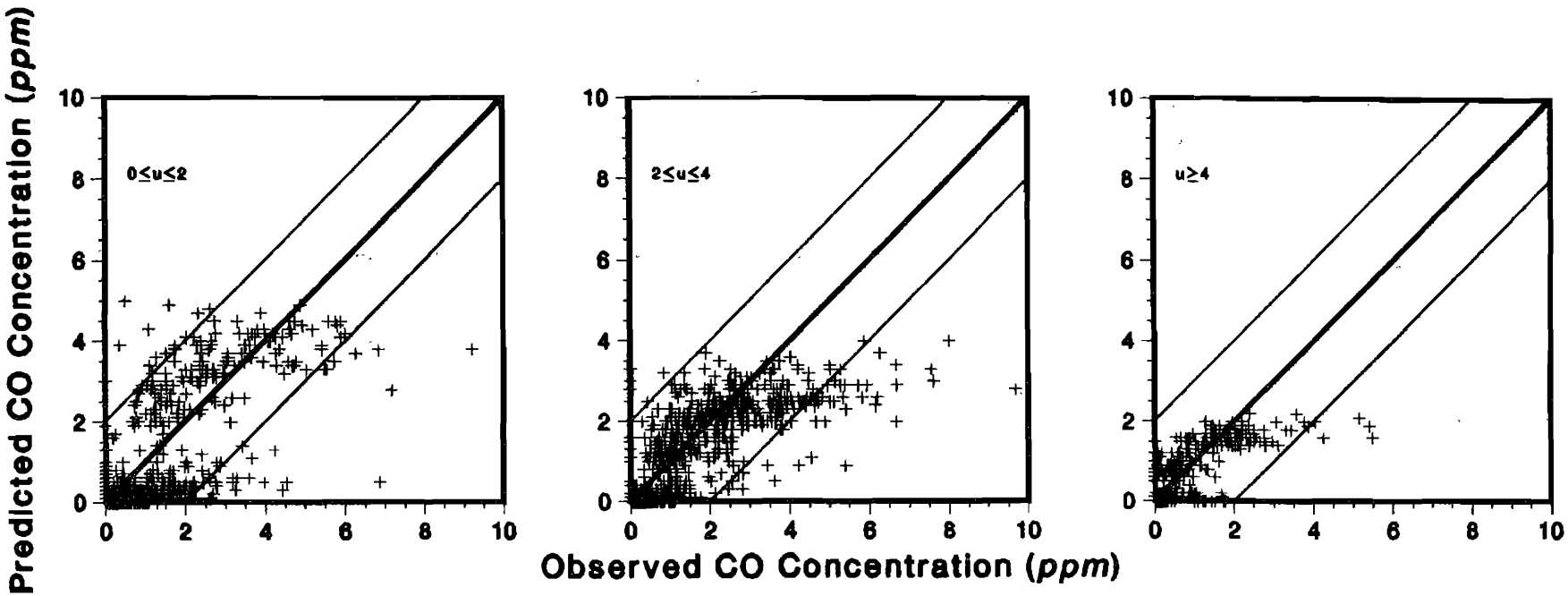
California Data Base



Scatterplots for Near 45° Wind Cases

TEXIN2 Operations & Design—Short Cut Method

California Data Base



Scatterplots for Near Perpendicular Wind Cases

TEXIN2 Operations & Design—Short Cut Method

California Data Base

Appendix G

Final Data Base Format Summary

Appendix G

A Detailed Description of the Data Base Format

The purpose of this Appendix is to present a detailed description of the structure of the experimental data base as it appears on magnetic tape. The data base is sorted so that all data for a given day is grouped together chronologically by ascending channel number. All instrument data are contained on one of three record formats: console records, radar records, and regular instrument channel records. The data base format presented in this appendix was generated by the SETD program contained in Appendix D. The maximum record length of each record was 80 bytes so that the data base was saved in card image format.

The first set of records for a given day are the console records. These records were issued by the operator while data was being acquired. These records contain information on instrument malfunctions, calibration starting and ending times, and other explanations that may be of importance when analyzing the data. The first data on each console record like all other records in the data base is a time stamp that indicates the 24 hour military time that the record was generated. The first two digits indicate the hours, the second two digits indicate the minutes, and the last two digits indicate the seconds. On those days where the data acquisition period began on one day and ended past midnight on the next day, the hour will be greater than 24. This is due to the fact that once the computer read the clock at the start of data acquisition, time was kept by the generation of internal interrupts. Therefore, the time was never reset at midnight. The next two numerical digits on the console records are zeros. The console message follows the two zeros on the record. A single record may contain up to 70 characters of the console message. If more characters are required, the message is continued on the next record. A FORTRAN program can read the console messages with the FORMAT statement: 3I2,1X,I2,1X,A70.

The next set of records for each day are the radar records. These records contain traffic data for a single radar for one minute. The first field on the radar records contains the time stamp as described with the console message records. The next field contains a two digit channel number for the radar. As indicated in Table 17 of this report, the radars are channels 1–10. Following the channel number is a record type used to determine the validity of the data on the record. Normal radar records have a type of 10, but if a radar record type is zero, the data are invalid on that record. There are five data fields on each radar record corresponding to the five vehicle categories designated by the computer. Each field contains two numbers. The first is the number of vehicles of that class passing under the radar for that minute. The second contains the average speed of the vehicles in that class. Radar classification schemes are discussed in Chapter 5. The radar data is ordered so that category 1 vehicles are presented first and the category 5 vehicles are presented last on each record. A FORTRAN program can read the radar data with the FORMAT statement:

3I2,1X,I2,1X,I2,1X,5(I5,1X,F5.0).

The final set of records for a each day are the regular channel records. This set contains the majority of the data. All instruments except the radars have data formats that fit this class. All samples from calibration instruments such as the Ecolyzers were adjusted for calibration drifts before recording in the final data base. Furthermore, the UVW corrections due to non-cosine response were also applied before the data were recorded. The first field on each of these records contains a time stamp corresponding to the time that the first sample on each record was obtained. The time stamp is followed by a two digit channel number as with the radars. Following the time stamp are six discrete instrument samples along with the record type of each channel at the time the sample was taken. Note that the record type is presented before each discrete sample. Normal run-time record types for each channel were presented in Table 17. A record type of zero means that the sample is invalid, while a record type of one or two indicates that the instrument was in the calibration state when the sample was taken. Record type 2 indicates a calibration span value while record type 1 indicates a calibration zero value. Since all data for a channel is stored chronologically, the sample rate for any channel can easily be determined by taking the difference between the time stamps on two adjacent records and dividing by six. All regular channel records can be read with the following FORTRAN FORMAT statement: 3I2,1X,I2,6(I3,1X,F7.0).

The final format of the data base consists of the above described types of records separated by terminating records. Furthermore, the first record for each date consists of a Julian date stamp. After the Julian date stamp are the console message records followed by a terminator. The terminator can easily be detected by the presence of a time stamp of 999999. After the console message records are the radar records. Each radar channel is preceded by a channel descriptor and followed by a terminator. The terminator may be read with the same format statement used to read the radar records. As with the console messages, the terminator contains a time stamp of 999999. The channel descriptor follows the same format for all channels including the regular channels and contains a time stamp of 000000 followed by the channel number which is followed by two channel numbers used to associate the channels of a UVW anemometer. For all instruments besides the UVW anemometers these associated channel numbers are 99. For the UVW anemometers these channel numbers are the channel numbers of the two other anemometers that are on the same mast. These channel numbers are followed by the normal channel record type, the channel name, and the units for the channel. A FORTRAN program should use the following FORMAT statement to read the channel description record: I6,4I3,5X,A12,5X,A8. Following the radar records are the regular channel records. Each regular channel is preceded by a channel description record and followed by a terminator record. The terminator records for the regular channels can be read with the same format statement used to read the records containing normal data. Following the last data records for a day are two terminator records. An example of the data base format is presented on the

following few pages.

```
1984.320
102658 00 Data run for day of 11/15/84.  Operators are Michael Hlavinka and Guy
102658 00 Donaldson.
102845 00 BEGIN CALIBRATION OF DASIBI CO        MWH
110550 00 END CALIBRATION OF DASIBI CO.
999999  0 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000  1 99 99 10         Radar 1          mph
102757  1 10      1  54.8    5  61.8    2  61.4    0    .0    0    .0
102857  1 10      7  57.3    7  60.6    1  72.3    1  66.2    0    .0
102957  1 10      3  59.4    6  66.1    2  66.1    1  67.4    0    .0
103057  1 10      4  57.4    7  62.9    0    .0    0    .0    1  56.4
103157  1 10      4  59.6    2  59.6    0    .0    0    .0    0    .0
999999  1 99     99 999.9   99 999.9   99 999.9   99 999.9   99 999.9
000000  2 99 99 10         Radar 2          mph
102757  2 10      1  61.9    8  64.5    2  63.8    1  62.5    0    .0
102857  2 10      4  62.2   11  60.2    1  67.6    0    .0    0    .0
102957  2 10      2  60.9    8  64.3    1  62.5    0    .0    0    .0
103057  2 10      2  63.5   15  62.1    0    .0    3  64.3    0    .0
103157  2 10      0    .0   12  62.3    0    .0    1  67.8    0    .0
999999  2 99     99 999.9   99 999.9   99 999.9   99 999.9   99 999.9
000000 11 13 15 19       N TOP PER          mph
102708 11 19    1.70 19    2.09 19    3.02 19    3.44 19    3.53 19    4.43
102808 11 19    9.60 19    5.25 19    6.67 19    9.25 19   11.41 19    7.99
102908 11 19    8.24 19    7.42 19   10.19 19    8.50 19    6.66 19    5.94
103008 11 19    4.66 19    4.09 19    1.05 19    3.74 19    6.94 19   11.70
103108 11 19    9.72 19    8.75 19    7.13 19    5.04 19    5.04 19    8.03
999999 11 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000 12 99 99  7       N Dasibi CO          ppm
110524 12  7    1.42  7    1.44  7    1.47  7    1.49  7    1.59  7    1.69
110536 12  7    1.56  7    1.61  7    1.59  7    1.59  7    1.61  7    1.61
110548 12  7    1.61  7    1.54  7    1.56  7    1.56  7    1.54  7    1.56
110600 12  7    1.49  7    1.44  7    1.39  7    1.42  7    1.37  7    1.44
110612 12  7    1.29  7    1.29  7    1.32  7    1.25  7    1.20  7    1.15
999999 12 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000 13 11 15 19       N TOP PAR          mph
102708 13 19    2.17 19    1.53 19   -1.56 19    -.97 19     .73 19    1.57
102808 13 19    -.36 19   -1.33 19    1.33 19    3.26 19    3.75 19    4.99
102908 13 19    2.42 19    1.09 19    5.50 19    2.06 19     .61 19    2.06
103008 13 19     .00 19    3.55 19   -1.00 19    1.09 19    -.24 19     .48
103108 13 19    1.82 19     .61 19     .00 19     .36 19     .48 19    5.32
999999 13 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000 14 99 99  3       N Dasibi O3          ppm
102718 14  3    .021  3    .022  3    .020  3    .020  3    .029  3    .025
102918 14  3    .020  3    .022  3    .021  3    .019  3    .018  3    .018
103118 14  3    .017  3    .017  3    .019  3    .023  3    .023  3    .028
103318 14  3    .027  3    .021  3    .016  3    .016  3    .017  3    .019
103518 14  3    .026  3    .026  3    .023  3    .022  3    .021  3    .023
999999 14 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000 15 11 13 19       N TOP VER          mph
102703 15 19     .25 19     .00 19     .50 19     .13 19    1.13 19    1.50
102733 15 19    1.00 19    1.25 19    1.38 19     .13 19     .00 19    1.00
102803 15 19    -.50 19   -1.00 19    2.13 19     .38 19     .25 19     .50
102833 15 19    1.63 19     .25 19    -.38 19    -.13 19   -2.25 19    1.38
102903 15 19    2.38 19    -.63 19    -.88 19   -1.50 19    -.88 19   -2.38
999999 15 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000 25 99 99 13       N WV 102'        DEGR AZ
102703 25 13  204.01 13  180.26 13  160.92 13  160.92 13  180.26 13  189.94
102733 25 13  196.97 13  163.56 13  188.18 13  188.18 13  192.57 13  192.57
102803 25 13  189.06 13  201.37 13  195.21 13  194.33 13  186.42 13  193.45
102833 25 13  198.73 13  196.09 13  196.97 13  196.97 13  193.45 13  191.70
102903 25 13  187.30 13  187.30 13  187.30 13  195.21 13  200.49 13  189.94
999999 25 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000 27 99 99 14       N Gnd Temp         DEGR F
102758 27 14   82.72 14   82.72 14   86.62 14   81.54 14   83.89 14   83.50
103358 27 14   85.45 14   85.06 14   87.80 14   82.72 14   83.11 14   84.67
103958 27 14   87.02 14   87.41 14   85.45 14   87.80 14   80.76 14   87.41
104558 27 14   85.45 14   81.54 14   85.84 14   80.76 14   77.64 14   82.33
```

```
105158 27 14    79.98 14    85.06 14    86.23 14    84.67 14    85.84 14    82.33
999999 27 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000 35 99 99 12        N HA 74'          mph
102708 35 12     5.86 12     6.35 12     4.89 12     9.77 12     9.04 12     8.79
102808 35 12     5.62 12    10.99 12     5.37 12    10.99 12     9.04 12    10.99
102908 35 12     6.84 12     7.82 12    13.19 12    10.50 12     9.77 12     8.79
103008 35 12     8.30 12     3.18 12     6.35 12     4.64 12     6.84 12    13.43
103108 35 12    12.21 12     8.30 12     7.33 12     4.15 12     7.33 12     9.77
999999 35 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000 51 99 99 17      N Ecolyzer-2        ppm
110824 51  1      .49  1      .37  1      .49  1      .37  1      .37  1      .24
110836 51  1      .37  1      .37  1      .37  1      .49  1      .49  1      .24
110848 51  1      .37  1      .37  1      .49  1      .37  1      .49  1      .37
110900 51  1      .37  1      .49  1      .37  1      .37  1      .37  1      .37
110912 51  1      .61  1      .37  1      .37  1      .37  1      .37  1      .37
110924 51  1      .37  1      .37  1      .24  1      .37  1      .37  1      .49
110936 51  0      .49  0      .49  0      .37  0      .37  0      .49  0      .37
110948 51  0      .37  0      .37  0      .37  0      .37  0      .37  0      .24
111000 51  0      .37  0      .61  0      .98  0     1.47  0     1.95  0     2.20
111012 51  0     2.69  0     3.79  0     5.37  0     6.96  0     8.18  0     9.40
111024 51  0    10.38  0    10.87  0    11.48  0    11.72  0    11.97  0    10.63
111036 51  0    10.26  0    10.50  0    10.63  0    10.63  0    10.63  0    10.50
111048 51  0    10.50  0    10.38  0    10.50  0    10.63  0    10.38  0    10.38
111100 51  0    10.50  0    10.38  0    10.38  0    10.38  0    10.50  0    10.50
111112 51  0    10.50  0    10.38  0    10.50  2    10.38  2    10.38  2    10.38
111124 51  2    10.50  2    10.50  2    10.38  2    10.38  2    10.50  2    10.50
111136 51  2    10.63  2    10.38  2    10.38  2    10.38  2    10.38  2    10.38
111148 51  2    10.38  2    10.38  2    10.38  2    10.38  2    10.38  2    10.38
111200 51  2    10.26  2    10.38  2    10.38  2    10.38  2    10.38  2    10.38
111212 51  2    10.38  2    10.38  2    10.38  2    10.38  2    10.38  2    10.38
111224 51  2    10.26  2    10.38  2    10.26  2    10.26  2    10.38  2    10.26
111236 51  2    10.26  2    10.38  2    10.26  0    10.26  0    10.38  0    10.38
999999 51 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000 59 99 99 15        N Rel Hum           %
102758 59 15    81.34 15    82.32 15    83.05 15    82.80 15    82.32 15    81.83
103358 59 15    80.36 15    80.85 15    82.07 15    80.12 15    79.38 15    77.92
103958 59 15    77.92 15    77.92 15    77.92 15    77.19 15    77.43 15    76.70
104558 59 15    76.45 15    76.45 15    76.21 15    76.94 15    77.43 15    76.45
105158 59 15    76.70 15    75.72 15    75.96 15    75.72 15    76.21 15    76.45
999999 59 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000 61 99 99 16      N Solar Rad       W/m**2
102758 61 16   255.47 16   222.08 16   229.58 16   660.80 16   661.48 16   675.11
103358 61 16   698.27 16   257.51 16   646.50 16   664.89 16   657.40 16   641.73
103958 61 16   636.28 16   639.00 16   626.06 16   630.83 16   632.87 16   639.69
104558 61 16   649.90 16   652.63 16   644.45 16   645.14 16   642.41 16   644.45
105158 61 16   646.50 16   649.22 16   657.40 16   647.18 16   653.31 16   654.67
999999 61 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
000000 64 99 99 18        S VA 55'          mph
102703 64 18     .67 18    -.48 18    -.96 18    2.49 18     .48 18    -.48
102733 64 18     .86 18    1.15 18    1.91 18     .38 18    -.86 18    -.96
102803 64 18    -.96 18    -.67 18    -.86 18    -.38 18    -.67 18    -.57
102833 64 18    -.19 18    -.67 18     .00 18     .00 18     .29 18     .48
102903 64 18    1.15 18     .77 18     .48 18    1.15 18    1.82 18     .86
999999 64 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
999999 99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99 99 9999.99
```

Appendix H

Sample Mass Balance Calculation

Appendix H

Sample Mass Balance Calculation

In order to illustrate the process of estimating emission factors by material balances, the following calculation is illustrated. The process involves plotting the flux of the species of interest versus receptor height and graphically integrating the resulting curve. The sample calculation below was taken from the data on December 7, 1984, during the five minute span from 2040–2045 CST. The data represents carbon monoxide concentrations.

A summary of the averaged data for this period is presented below.

Height (ft)	Wind Speed (mph)	Wind Direction (deg)	Concentration (ppm)
102	6.5	194	1.67
43	3.4	205	2.07
33	1.5	200	2.87
5	1.6	183	2.41

The above data include information on four downwind Ecolyzers. At the time, one of the downwind Ecolyzers was not in service. The traffic data for the same time period may be summarized as follows:

Category	Total Vehicles	Average Speed (mph)	Percent Load
1	156	57.2	40.5
2	215	58.9	55.8
3	9	61.9	2.3
4	5	55.2	1.3
5	0	0.0	0.0

Furthermore, the background concentration observed during the interval was 1.78 ppm. This value was taken as the average of all functional upwind CO monitors.

The first step in the calculation procedure involves determining the cross wind component of the wind speed. This value is determined by:

$$u_x = u|\sin(\theta - \theta_0)| \qquad (H-1)$$

where:

u = average wind speed

θ = angle of wind direction

θ_0 = compass heading of roadway with respect to north

All wind instruments were directed such that 0° referred to a line running perpendicular to the freeway pointing north. Therefore, the effective value of θ_0 was 90°.

After determining u_x, the mass flux of pollutant flowing across the roadway is calculated. The background concentration is first subtracted from the downwind receptor concentrations to determine the amount of pollutant generated from the traffic. If this value is negative, the effective concentration is set to zero. The concentrations are then converted from a volume basis to a mass basis. At conditions in the atmosphere, the ideal gas law may be used to calculate gas density:

$$\rho = \frac{PM}{RT} \qquad (H-2)$$

where:

$P =$ atmospheric pressure

$R =$ ideal gas constant $= 8.205 \times 10^{-5} \frac{m^3 \cdot atm}{gmole \cdot K}$

$T =$ absolute temperature

$M =$ molecular weight of the gas of interest

For example, using the concentration of CO measured at 5 feet, the mass flux may be determined as follows:

$$\text{Effective concentration} = 2.41 - 1.78 = 0.63\,\text{ppm}$$

At an ambient temperature of 47°F (281.7K) and atmospheric pressure,

$$\rho_{CO} = 1121 \frac{\text{gm CO}}{\text{m}^3}$$

since the molecular weight of CO is

$$M_{CO} = 28 \frac{\text{gm CO}}{\text{gmole}}$$

From the definition of parts per million,

$$1\,\text{ppm CO} = 1 \frac{\text{m}^3\,\text{CO}}{10^6\,\text{m}^3_{\text{air}}}$$

Therefore, one ppm CO is equivalent to 1121 gm CO in $10^6\,\text{m}^3$ of air, or

$$C_{CO} = 0.63\,\text{ppm} \left(\frac{1121\,\text{gm CO}}{10^6\,\text{m}^3_{\text{air}} \cdot \text{ppm}} \right)$$

or the mass concentration of CO is,

$$C_{CO} = 7.63 \times 10^{-4} \frac{\text{gm CO}}{\text{m}^3_{\text{air}}}$$

Using equation (H-1),

$$u_x = u|\sin(\theta - \theta_0)| = 1.6|\sin(183° - 90°)|$$

$$u_x = 0.7\,\mathrm{m/sec}$$

The mass flux, G, is then given by:

$$G_{CO} = u_x C_{CO} = (0.7\,\mathrm{m/sec})(7.63 \times 10^{-4}\,\frac{\mathrm{gm\,CO}}{\mathrm{m^3_{air}}})(3600\,\mathrm{sec/hr})$$

or,

$$G_{CO} = 2.0\,\frac{\mathrm{gm\,CO}}{\mathrm{m^2 \cdot hr}}$$

Performing the same calculations on all receptors, yields the following table:

Receptor Height (m)	u_x (m/sec)	G_{CO} $\left(\frac{\mathrm{gm\,CO}}{\mathrm{m^2 \cdot hr}}\right)$
31.1	2.8	0.0
14.3	1.4	1.73
10.1	0.6	2.95
1.5	0.7	2.00

If the mass flux is plotted against receptor height and the area below the resulting curve determined from graphical integration, the composite emission factor is obtained as follows:

$$\text{Area below curve} = 46.5\,\frac{\mathrm{gm\,CO}}{\mathrm{m \cdot hr}}$$

$$\text{Total \# vehicles} = 156 + 215 + 9 + 5 = 385$$

$$\text{Emission Factor} = \left(46.5\,\frac{\mathrm{gm\,CO}}{\mathrm{m \cdot hr}}\right)\left(\frac{5\,\mathrm{min}}{385\,\mathrm{vehicles}}\right)\left(\frac{\mathrm{hr}}{60\,\mathrm{min}}\right)\left(\frac{1609\,\mathrm{m}}{\mathrm{mi}}\right)$$

or,

$$\text{Emission Factor} = 16.2\,\frac{\mathrm{gm\,CO}}{\mathrm{vehicle \cdot mi}}$$

Figure H1 illustrates the mass flux profile obtained from this five minute period.

The same process was used to compare the measured SF_6 emission rates to the observed rates. An example calculation for SF_6 is given below.

The period chosen for the demonstration was the 15 minute span from 1408–1423 on December 19, 1984. A 15 minute interval was used because the sample length for a single SF_6 syringe was 15 minutes. The meteorology and observed concentrations are given in Appendix I and restated here for convenience.

Estimation of Vehicular CO Emission Factors

CO Mass Flux vs. Receptor Height

Houston, Texas

IH610 between Airline Dr. & N. Main St.

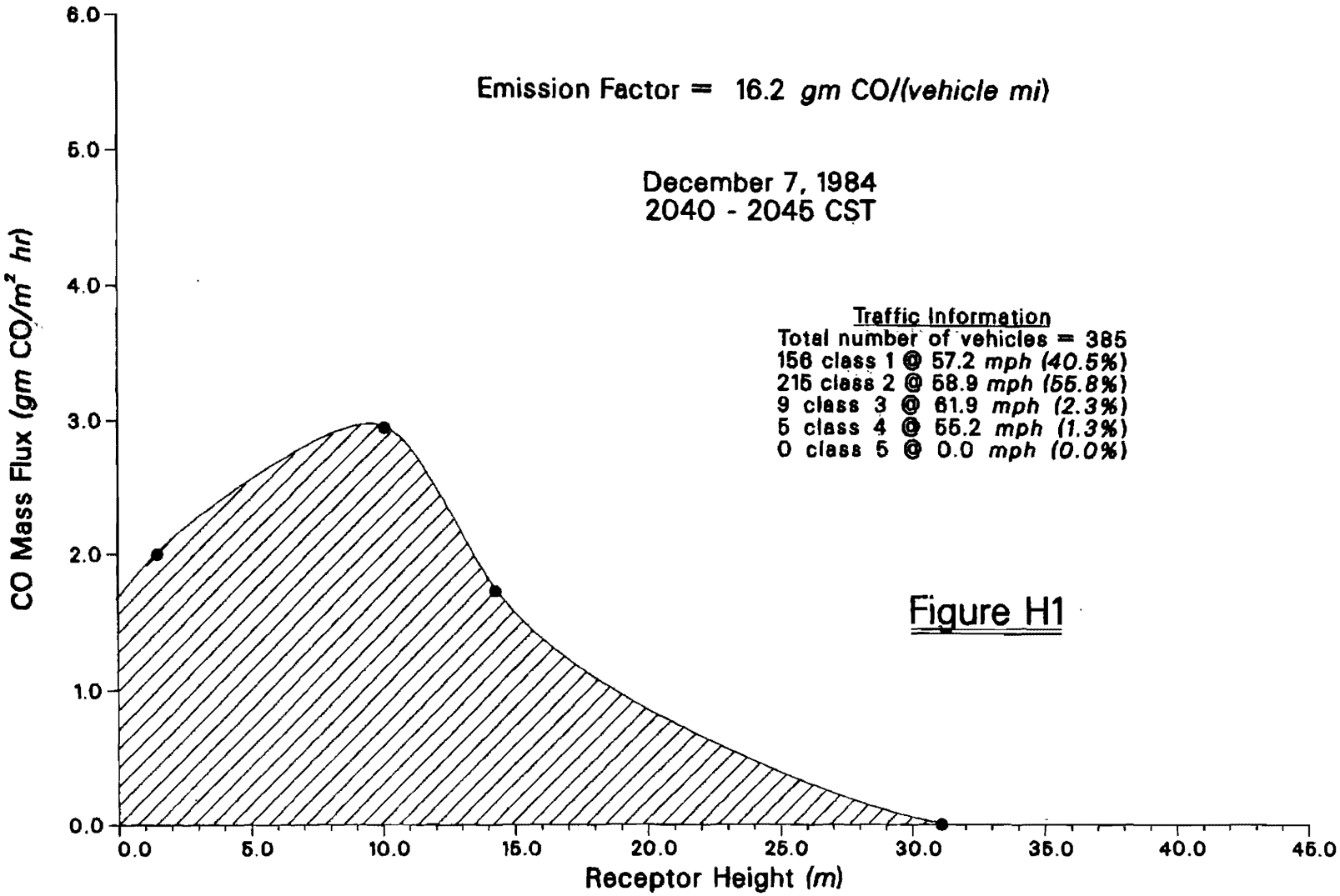


Figure H1

Height (ft)	Wind Speed (mph)	Wind Direction (deg)	Concentration (ppb)
59	4.9	193	0.722
42	4.3	192	1.426
25	3.6	176	2.571
5	3.2	153	3.413

Additionally, the distance between N. Main and Airline is about 883 m. The ambient temperature was about 76°F. Each of the two vehicles dispersing tracer passed the downwind sampling tower 15 times during the interval.

The cross wind component speeds are again calculated using equation (H–1). Next, the mass flux of SF_6, at each receptor height, moving across the roadway is determined. At the 42 ft receptor, the measured SF_6 concentration was 1.426 ppb. With an average upwind concentration of 0.429 ppb, the effective observed concentration is:

$$\text{Effective observed concentration} = 1.426 - 0.429 = 0.997\,\text{ppb}$$

At the stated temperature and a pressure of 1 atm, the density of SF_6 is:

$$\rho_{SF_6} = \frac{PM_{SF_6}}{RT}$$

and since,

$$M_{SF_6} = 146\,\frac{\text{gm SF}_6}{\text{gmole}}$$

the density is,

$$\rho_{SF_6} = 5982\,\frac{\text{gm SF}_6}{\text{m}^3}$$

Since,

$$1\,\text{ppb} = \frac{1\,\text{m}^3\,\text{SF}_6}{10^9\,\text{m}^3_{\text{air}}}$$

the observed concentration and the density may be used to calculate the concentration of tracer on a mass basis,

$$C_{SF_6} = 0.997\,\text{ppb}\left(\frac{5982\,\text{gm SF}_6}{10^9\,\text{m}^3_{\text{air}}\cdot\text{ppb}}\right)$$

or,

$$C_{SF_6} = 5.96\times10^{-6}\,\frac{\text{gm SF}_6}{\text{m}^3}$$

Using (H–1),

$$u_x = u|\sin(\theta-\theta_0)| = 4.3|\sin(192^\circ-90^\circ)|$$

$$u_x = 4.2\,\text{mph} = 1.88\,\text{m/sec}$$

Thus the mass flux of SF_6, G_{SF_6} is:

$$G_{SF_6} = u_x C_{SF_6} = 1.88\,\text{m/sec}\left(5.96 \times 10^{-6}\,\frac{\text{gm SF}_6}{\text{m}^3}\right)\left(3600\,\frac{\text{sec}}{\text{hr}}\right)\left(\frac{1000\,\text{mg}}{\text{gm}}\right)$$

or,

$$G_{SF_6} = 40.7\,\frac{\text{mg SF}_6}{\text{m}^2 \cdot \text{hr}}$$

Performing the above calculations for each receptor yields the following table:

Height (m)	u_x (m/sec)	$G_{SF_6}\left(\frac{\text{mg SF}_6}{\text{m}^2\cdot\text{hr}}\right)$
18.0	2.1	13.35
12.8	1.9	40.77
7.6	1.6	72.21
1.5	1.3	81.53

Plotting the above data for SF_6 mass flux *vs.* receptor height, and graphically integrating the resulting curve to determine the area below the curve gives an emission rate (on a linear basis) of $1054\,\frac{\text{mg SF}_6}{\text{m}\cdot\text{hr}}$. Since the length of the road where the tracer was emitted was 883 meters, the observed emission rate is then:

$$\text{Observed emission rate} = 1054\,\frac{\text{mg SF}_6}{\text{m} \cdot \text{hr}}\left(\frac{\text{hr}}{60\,\text{min}}\right)\left(\frac{\text{gm}}{1000\,\text{mg}}\right)(883\,\text{m})$$

or,

$$\text{Observed emission rate} = 15.5\,\frac{\text{gm SF}_6}{\text{min}}$$

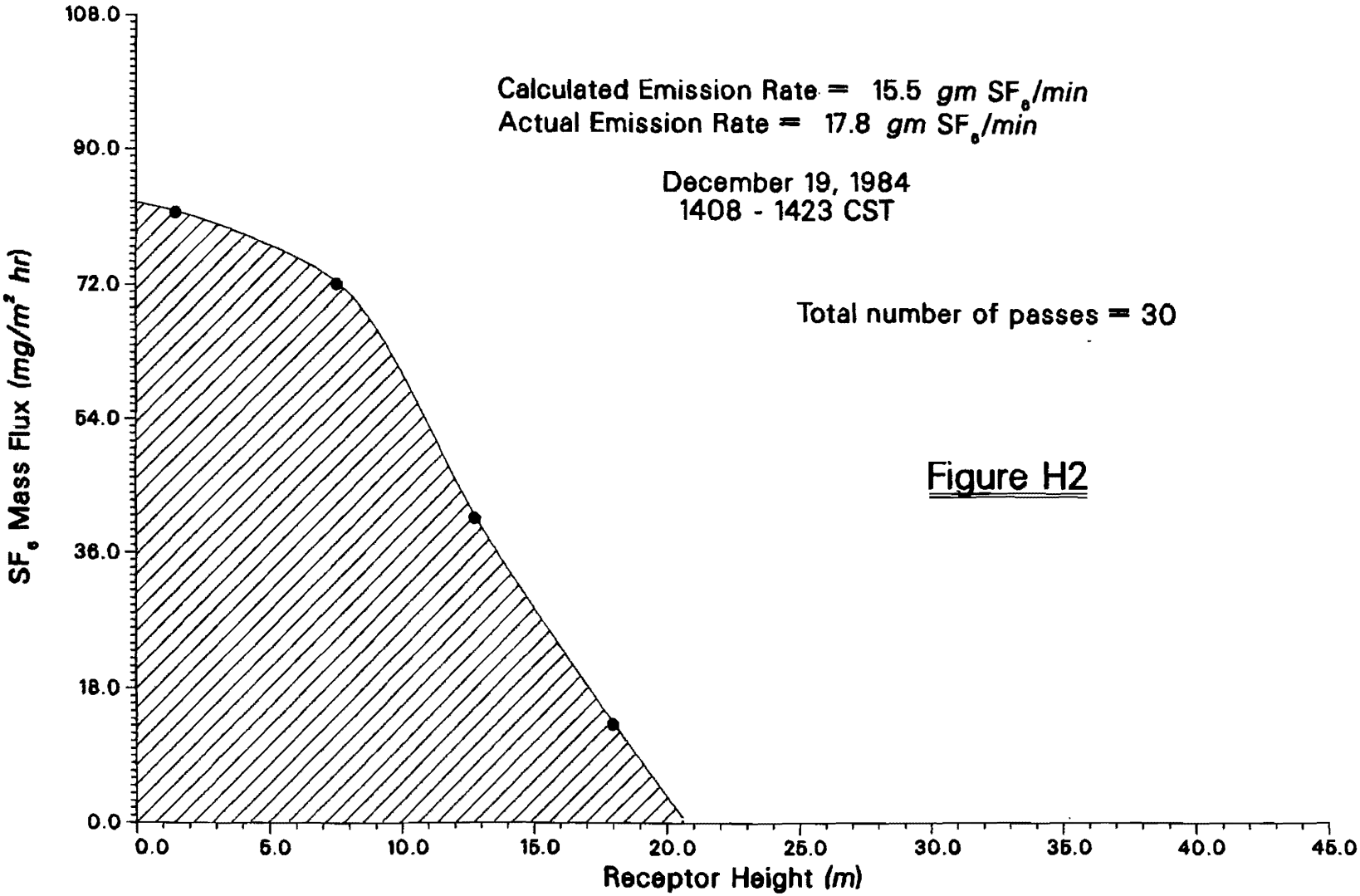
Figure H2 gives the plot that was used in the graphical integration.

Comparison of SF_6 Tracer Emission Factors

SF_6 Mass Flux vs. Receptor Height

Houston, Texas

IH610 between Airline Dr. & N. Main St.



Calculated Emission Rate = 15.5 *gm* SF_6/*min*
Actual Emission Rate = 17.8 *gm* SF_6/*min*

December 19, 1984
1408 - 1423 CST

Total number of passes = 30

Figure H2

Appendix I

SF_6 Profile Tables

12/18/84
1309–1324 CST

Temperature = 22.3°C
SF_6 density = 6023 gm/m^3
Background Concentration = 0.003 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
47	14.3	8.0	116	0.123
33	10.1	7.8	109	0.200
16	4.9	7.4	102	0.280
5	1.5	7.2	105	0.482

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
14.3	1.6	4.1
10.1	1.1	4.9
4.9	0.7	4.2
1.5	0.9	9.2

12/18/84
1324–1339 CST

Temperature = 22.7°C
SF_6 density = 6015 gm/m^3
Background Concentration = 0.005 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
47	14.3	9.1	128	0.036
33	10.1	8.2	121	0.115
5	1.5	7.5	116	0.512

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
14.3	2.5	1.7
10.1	1.9	4.5
1.5	1.5	16.7

12/18/84
1339–1354 CST

Temperature = 23.6°C
SF_6 density = 5996 gm/m^3
Background Concentration = 0.002 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
47	14.3	8.3	135	0.064
16	4.9	6.5	116	0.118
5	1.5	6.6	116	0.326

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
14.3	2.6	3.5
4.9	1.3	3.2
1.5	1.3	9.1

12/18/84
1354–1409 CST

Temperature = 23.1°C
SF_6 density = 6005 gm/m^3
Background Concentration = 0.004 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
47	14.3	9.2	121	0.059
33	10.1	8.8	112	0.092
16	4.9	8.4	102	0.199
5	1.5	8.3	107	0.352

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
14.3	2.1	2.5
10.1	1.5	2.9
4.9	0.8	3.5
1.5	1.1	8.2

12/18/84
1409–1424 CST

Temperature = 23.1°C
SF_6 density = 6006 gm/m^3
Background Concentration = 0.003 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
47	14.3	7.7	122	0.081
16	4.9	7.1	104	0.293
5	1.5	7.0	108	0.515

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
14.3	1.8	3.1
4.9	0.8	5.1
1.5	1.0	10.9

12/18/84
1424–1439 CST

Temperature = 23.8°C
SF_6 density = 5992 gm/m^3
Background Concentration = 0.003 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
47	14.3	7.5	121	0.225
33	10.1	7.4	110	0.255
16	4.9	7.2	101	0.358
5	1.5	7.1	109	0.498

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
14.3	1.7	8.3
10.1	1.2	6.3
4.9	0.6	4.8
1.5	1.0	11.1

12/19/84
1107–1122 CST

Temperature = 22.4°C
SF_6 density = 6020 gm/m^3
Background Concentration = 0.809 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	4.9	210	0.946
42	12.8	4.5	213	0.454
5	1.5	2.6	189	2.199

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	1.9	5.7
12.8	1.7	0.0
1.5	1.2	34.9

12/19/84
1122–1137 CST

Temperature = 22.9°C
SF_6 density = 6010 gm/m^3
Background Concentration = 1.206 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	4.4	193	1.483
42	12.8	3.9	193	1.534
25	7.6	3.2	182	2.020
5	1.5	2.8	169	3.631

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	1.9	11.5
12.8	1.7	12.0
7.6	1.4	25.2
1.5	1.2	63.8

12/19/84
1137–1152 CST

Temperature = 23.9°C
SF_6 density = 5991 gm/m^3
Background Concentration = 1.227 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	5.2	178	1.364
42	12.8	4.5	176	1.815
25	7.6	3.7	165	1.798
5	1.5	3.8	146	3.171

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	2.3	6.8
12.8	2.0	25.7
7.6	1.6	19.9
1.5	1.4	59.1

12/19/84
1152–1207 CST

Temperature = 24.0°C
SF_6 density = 5988 gm/m^3
Background Concentration = 1.170 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	5.4	182	1.466
42	12.8	4.5	183	1.892
25	7.6	3.1	176	2.190
5	1.5	3.0	154	1.713

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	2.4	15.3
12.8	2.0	31.2
7.6	1.4	30.2
1.5	1.2	14.2

12/19/84
1207–1222 CST

Temperature = 24.6°C
SF_6 density = 5975 gm/m^3
Background Concentration = 1.146 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	7.3	182	1.364
42	12.8	6.3	182	1.474
25	7.6	4.8	171	2.506
5	1.5	3.7	154	2.949

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	3.2	15.2
12.8	2.8	20.0
7.6	2.1	61.9
1.5	1.5	57.7

12/19/84
1222–1237 CST

Temperature = 24.5°C
SF_6 density = 5979 gm/m^3
Background Concentration = 1.834 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	6.1	189	1.219
42	12.8	5.1	194	1.867
25	7.6	3.5	190	2.659
5	1.5	3.2	158	3.392

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	2.7	0.0
12.8	2.2	1.6
7.6	1.5	27.2
1.5	1.3	44.3

12/19/84
1408–1423 CST

Temperature = 24.3°C
SF_6 density = 5982 gm/m^3
Background Concentration = 0.429 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	4.9	193	0.722
42	12.8	4.3	192	1.426
25	7.6	3.6	176	2.511
5	1.5	3.2	153	3.413

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	2.1	13.4
12.8	1.9	40.8
7.6	1.6	72.2
1.5	1.3	81.5

12/19/84
1423–1438 CST

Temperature = 24.9°C
SF_6 density = 5970 gm/m^3
Background Concentration = 0.537 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	5.3	175	1.155
42	12.8	4.7	174	1.995
25	7.6	3.8	160	2.765
5	1.5	3.5	141	2.678

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	2.4	31.3
12.8	2.1	64.9
7.6	1.6	75.8
1.5	1.2	57.0

12/19/84
1438–1453 CST

Temperature = 24.9°C
SF_6 density = 5969 gm/m^3
Background Concentration = 0.407 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	5.4	172	2.083
42	12.8	4.8	168	2.310
25	7.6	3.8	153	2.450
5	1.5	3.7	130	3.938

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	2.4	86.8
12.8	2.1	86.1
7.6	1.5	66.7
1.5	1.1	81.5

12/19/84
1453–1508 CST

Temperature = 24.6°C
SF_6 density = 5977 gm/m^3
Background Concentration = 0.326 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	5.4	141	2.258
42	12.8	5.0	139	2.380
25	7.6	4.5	128	3.325
5	1.5	4.9	114	3.448

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	1.9	78.5
12.8	1.7	75.6
7.6	1.2	80.2
1.5	0.9	62.1

12/19/84
1508–1523 CST

Temperature = 25.1°C
SF_6 density = 5966 gm/m^3
Background Concentration = 0.451 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	5.9	169	2.730
42	12.8	5.2	167	2.144
25	7.6	4.1	157	2.371
5	1.5	3.9	134	1.470

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	2.6	126.6
12.8	2.3	82.5
7.6	1.7	70.3
1.5	1.2	26.6

12/19/84
1523–1538 CST

Temperature = 25.0°C
SF_6 density = 5968 gm/m^3
Background Concentration = 0.812 ppb

General Information

Z (ft)	Z (m)	u (mph)	θ	Conc (ppb)
59	18.0	5.6	197	1.588
42	12.8	4.9	201	1.715
25	7.6	3.6	194	0.827
5	1.5	2.9	165	0.459

Mass Flux Data

Z (m)	u_x (m/sec)	G (mg/m^2 · hr)
18.0	2.4	40.1
12.8	2.0	39.7
7.6	1.6	0.5
1.5	1.2	0.0

Appendix J

Aerial View of Houston Intersection Site