APPENDIX D: OSM AND SUMMARY OF OFF-ROAD EMISSIONS INVENTORY UPDATE

I. EMISSIONS INVENTORY DEVELOPMENT FOR IN-USE OFF-ROAD EQUIPMENT

A. Overview

Since the emissions inventory was developed in support of the 2007 Rulemaking, conditions in the construction industry have changed dramatically. The economic recession that technically ended in June 2009 was the most severe since the Great Depression, and had a devastating impact on the construction industry in California. Staff estimates that between 2005 and 2010 construction emissions dropped by more than 50% because of reduced demand for construction services caused by the recession. Other emissions categories, like use of industrial off-road equipment, oil-drilling equipment, and aircraft ground support equipment saw similar if less severe declines in emissions over the same period that were caused by the recession. As a result of the recession, the Board directed staff to develop rule amendments to provide regulatory relief.

In 2009 industry stakeholders pointed out a 2009 study by Rob Harley at UC Berkeley which used a fuel-based method to assess construction equipment emissions and found our inventory was overestimated by more than a factor of three. Industry also pointed out that a similar study focused on all off-road equipment nationally was published in 2000 by Kean, Sawyer and Harley that found similar results.

In light of this new information, we conducted a comprehensive review of the inventory. We evaluated new sources of information that were not available when the Rule was developed, and updated the emissions inventory to reflect these new data as well as the impact of the recession. We made several updates:

- Population

We updated our estimates using the equipment population reported to ARB for Rule compliance. The total population in 2009 was 26% lower than had been anticipated in 2007 due to fleet downsizing during the recession.

- Hours of Use

The Rule allowed fleets to report 2007 and 2009 activity to obtain credits towards meeting regulatory obligations. About 10% of regulated vehicles reported this information. 2007 was a relatively average year for construction so we expected data to be consistent with previous estimates. Instead new data by equipment type were in most cases at least 30% lower than we expected. 2009 activity was lower than 2007 due to the recession.

- Load Factor

New data from 2009 academic studies and from engine manufacturers suggested our load factors should be reduced by 33%.

- Growth Forecasts

Between 2005 and today construction activity and emissions have dropped by more than 50% based on many different indicators. Future emissions are uncertain and depend on the pace of economic recovery. To forecast emissions we reviewed a variety of economic forecasts, developed bounding scenarios to estimate the range of potential future emissions, and assumed the average of those forecast scenarios. Our analysis suggests off-road activity and emissions will recover from the recession slowly.

The major data sources used to develop these updates include: ARB's Diesel Off-road On-line Reporting System (DOORS) vehicle database; Equipment Data Associates (EDA) UCC1 filings; Assembly Bill 8 2x (AB 8 2x) reporting database; Eastern Research Group (ERG); Bureau of Economic Analysis (BEA); Bureau of Labor Statistics (BLS); Bureau of Transportation Statistics (BTS); U.S Environmental Protection Agency; and additional input gathered from stakeholders.

Our revised emissions are substantially lower than previously estimated. About half of this reduction can be attributed to the recession, and about half to updated assumptions independent of the recession. Table 1 compares the 2007 and updated inventories. Table 2 compares estimated fuel use in the 2007 inventory to the current inventory and industry estimates.

Table D-1: Comparison of 2007 Rule and Current Emission Estimates

Calendar Year	2007 Inventory (tons/day)		Current Inventory (tons/day)	
	PM2.5	**NOx**	**PM2.5**	**NOx**
2009	18.5	358	3.7	79
2014	13.1	272	3.6	76
2023	5.1	136	2.2	52

Table D-2: 2009 Estimated Fuel Use (Million Gallons of Diesel Fuel): ARB vs. Industry

2007 Inventory	Current Inventory	Industry Estimates
990	219	160 to 186

In this document, ARB staff describes the methodology used to estimate emissions from equipment subject to the proposed off-road diesel regulation. These include mobile (self-propelled) vehicles with engines greater than or equal to 25 horsepower. They are found in four categories within ARB's off-road diesel inventory: airport ground support equipment (GSE), construction and mining, industrial, and oil drilling. Table 3 shows the specific equipment covered by the proposed rule in each of these four categories.

Despite the major economic recession and revisions to the off-road Rule inventory, the in-use diesel off-road equipment category remains a substantial source of emissions. Staff estimates the off-road equipment subject to the Rule is the 6th largest source of diesel PM in California (7% of total) and the 8th largest source of NOx from all sources (4% of total). Emissions levels from in-use construction equipment are comparable to many other previously regulated sources including commercial harborcraft, locomotives, and cargo handling equipment.

Table D-3: Categories and Equipment Types Included in the Emissions Inventory

Category	Equipment
Airport Ground Support Equipment (GSE)	A/C Tug Narrow Body
	A/C Tug Wide Body
	Baggage Tug
	Belt Loader
	Bobtail
	Cargo Loader
	Cargo Tractor
	Forklift (GSE)
	Lift
	Other GSE
	Passenger Stand
Construction and Mining	Bore/Drill Rigs
	Cranes
	Crawler Tractors
	Excavators
	Graders
	Off-Highway Tractors
	Off-Highway Trucks
	Other Construction Equipment
	Pavers
	Paving Equipment
	Rollers
	Rough Terrain Forklifts
	Rubber Tired Dozers
	Rubber Tired Loaders
	Scrapers
	Skid Steer Loaders
	Surfacing Equipment
	Tractors/Loaders/Backhoes
	Trenchers
Industrial	Aerial Lifts
	Forklifts
	Other General Industrial Equipment
	Other Material Handling Equipment
	Sweepers/Scrubbers
Oil Drilling	Drill Rig (Mobile)
	Workover Rig (Mobile)

B. Methodology for Estimating Emissions

The PM and NOx emissions from each type of off-road rule equipment are calculated using the following equation:

$$\textit{Emissions in tons/day} = Pop * HP_{ave} * LF * Activity * EF$$

Where:

Pop	= Population
HP_{ave}	= Maximum rated average horsepower (hp)
LF	= load factor, unitless
Activity	= Activity or annual operation (hr/yr)
EF	= Emission factor (g/hp-hr)

The above equation is applied to a diverse population of vehicles and is extrapolated into the future. Two models were developed to accomplish this task: the Off-road Simulation Model (OSM) used current population characteristics and assumptions about vehicle turnover and purchasing age to estimate what those characteristics will be like in future years for a ‘baseline’ scenario (business-as-usual) and a ‘with-rule’ scenario (proposed rule effects on population); the Off-road Emissions Inventory model applies the above equation to the population generated by OSM using appropriate factors (EF, LF, Activity) for individual types of equipment. A more detailed description of these models and their coding are available in Attachment A. The data inputs are described in more detail below.

C. Emissions Inventory Inputs

1. Population

The off-road rule requires all owners of regulated equipment (Table 3) to report fleet information in ARB’s Diesel Off-road On-line Reporting System (DOORS). The total population of equipment subject to the off-road rule was based on DOORS data and an estimate for the number of vehicles that have not reported (see non-compliance section below).

a. DOORS

The DOORS database is an on-line tool designed to help fleet owners report their vehicle inventories for compliance with the off-road diesel regulation. It is also used by ARB to track compliance of individual fleets with the requirements of the regulation. Fleet owners were required to report all relevant fleet information including total number of vehicles, vehicle types, and vehicle model years. These data served as the basis for developing a statewide population inventory by vehicle type and model year/age. The final reporting date for fleet owners depended on the size of the fleet:

- Large fleets (fleets over 5,000 hp and State or government fleets): January 1, 2010
- Medium fleets (fleets between 2,501 and 5,000 hp): January 1, 2010
- Small fleets (fleets under 2,500 and other specific categories): March 1, 2010

For the purposes of emission inventory development, staff used a snapshot of the DOORS database from April 2010 as a base estimate of the 2009 population. Although all fleets should have reported to DOORS by that point in time, a fraction had not. As a result, it was necessary to determine the rate of non-compliance and make an adjustment to the DOORS population.

b. Non-Compliance Estimate

Staff conducted a survey of 1,000 potential fleet owners in March 2010 to estimate the number of vehicles subject to the off-road rule that had not reported to DOORS. This sample was developed from a list of 21,800 buyers who had financed construction equipment in 2005 to 2009 based on a collection of UCC1 forms filed with the California Secretary of State and subsequently assembled by Equipment Data Associates (EDA, 2010). UCC1 forms are financing statements supplied by lenders to the California Secretary of State to indicate a security interest in financed property. In order to avoid contacting fleet owners that had already reported to DOORS, staff filtered the sampling list to remove fleets in the DOORS database by matching company names and addresses. The final list of 1,000 fleet owners was created using a random number generator and stratified among different air basins to achieve a representative sample of the entire California population. Large, medium and small fleets were identified by the total value of equipment financed in the UCC1 database. Specifically, large fleets were those fleets with the highest total equipment values, followed by medium and small fleets. The final sample fleet characteristics are shown in Table 4. The air basin abbreviations are defined in Table 9.

Table D-4: Sample Fleet Characteristics (number of fleets)

		Air Basin						
		OTHER	SCAB	SDAB	SFBAB	SJVAB	SVAB	Total
Fleet Size	Large	110	128	31	63	107	61	500
	Medium	60	55	13	29	63	30	250
	Small	64	66	17	36	39	28	250
	Total	234	249	61	128	209	119	1,000

Of the 1,000 fleets contacted, 73 (about 7%) owned vehicles subject to reporting requirements that were not yet reported in DOORS. The other 927 fleets either could not be reached and therefore were assumed to no longer operate in California or were reached and confirmed that their off-road vehicles were not subject to the off-road rule (e.g. agricultural vehicles). Attachment B provides a summary of the survey results.

Staff applied this non-compliance rate to the 21,800 unique buyer fleets. Based on assumptions about fleet size (i.e. number of vehicles/fleet) for small, medium, and large fleets, staff estimated there were about 13,000 unreported vehicles statewide. The fleet size assumptions were based on an examination of the average fleet sizes reported to DOORS during the months of February through June 2010:

- Large fleet size: 25 vehicles per fleet
- Medium fleet size: 16 vehicles per fleet
- Small fleet size: 6 vehicles per fleet

The total estimated number of unreported vehicles equaled about 10% of the total DOORS population in April 2010 (about 130,000). As a result, staff applied a 10% adjustment factor to each of the equipment-specific populations to increase the total estimated DOORS population to about 144,000 vehicles. The original DOORS and final adjusted populations for each equipment type are shown in Table 5.

Table D-5: Original DOORS and Adjusted Equipment Populations

Category	Equipment	DOORS Population	Adjusted Population
Airport Ground Support Equipment (GSE)	A/C Tug Narrow Body	277	305
	A/C Tug Wide Body	131	144
	Baggage Tug	197	217
	Belt Loader	170	187
	Bobtail	35	39
	Cargo Loader	374	411
	Cargo Tractor	248	273
	Forklift (GSE)	271	298
	Lift	162	178
	Other GSE	879	967
	Passenger Stand	55	61
Construction and Mining	Bore/Drill Rigs	1,030	1,133
	Cranes	3,102	3,412
	Crawler Tractors	5,714	6,285
	Excavators	11,112	12,223
	Graders	3,593	3,952
	Off-Highway Tractors	2,528	2,781
	Off-Highway Trucks	2,646	2,911
	Other Construction Equipment	3,827	4,210
	Pavers	1,359	1,495
	Paving Equipment	770	847
	Rollers	7,078	7,786
	Rough Terrain Forklifts	7,137	7,851
	Rubber Tired Dozers	631	694
	Rubber Tired Loaders	9,250	10,175
	Scrapers	5,195	5,715
	Skid Steer Loaders	9,255	10,181
	Surfacing Equipment	425	468
	Tractors/Loaders/Backhoes	27,913	30,704
	Trenchers	1,677	1,845
Industrial	Aerial Lifts	6,286	6,915
	Forklifts	11,790	12,969
	Other General Industrial Equipment	3,002	3,302
	Other Material Handling Equipment	903	993
	Sweepers/Scrubbers	1,147	1,262
Oil Drilling	Drill Rig (Mobile)	167	184
	Workover Rig (Mobile)	580	638
Total		130,916	144,011

2. Engine Load Factor

The load factor is the average operational level of an engine in a given application as a fraction or percentage of the engine manufacturer's maximum rated horsepower. Since emissions are directly proportional to engine horsepower, load factors are used in the emissions inventory calculations to adjust the maximum rated horsepower to normal operating levels.

Load factors are difficult to characterize since they are highly dependent on equipment use and operation. The load factors in the original 2007 rulemaking were obtained from the Power Systems Research (1996) database. However, data collected in 2009 and 2010 suggest these original load factors were too high. The new data sources included an Eastern Research Group (ERG) study (ARB, 2008), ARB Mobile Source Control Division (MSCD) testing data (ARB, 2010), and data provided by manufacturers and the U.S. EPA (EPA, 2004a). The ARB data are discussed in Attachment C.

Some newer off-road diesel engines are equipped with an electronic control unit (ECU), which provides real-time load information. Additionally, other surrogates are used to estimate the engine load when direct load data is not available. These surrogates include manifold absolute pressure (MAP), exhaust temperature (Temp), and engine RPM (RPM). To use these surrogates to derive load, it is assumed that the engine is fully loaded when the surrogate reaches its maximum value during a duty cycle and the engine is unloaded when the surrogate is at its minimum value. The general formula to calculate load based on these surrogates is:

$$LF_t = \frac{X_t - X_{\min}}{X_{\max} - X_{\min}}$$

Where:

LF_t = Load factor at time t
X_t = Surrogate variable at time t, such as MAP, Temp, and RPM
X_{max} = Maximum surrogate value during a given duty cycle
X_{min} = Minimum surrogate value during a given duty cycle

After evaluating data from a variety of surrogates, staff concluded that electronic control unit (ECU) data provides better estimates of engine load than the other surrogates (ARB, 2010). Table 6 summarizes the load factor results from ECU data for tractors, forklifts, dozers and excavators. Also shown in Table 6 are the load factors from ARB's OFFROAD model, which was used to calculate the emissions inventory for the original rulemaking.

Table D-6: Electronic Control Unit (ECU) Load Factor Data

Equipment	HP Range	Model Year	OFFROAD Load Factor	Load Factor Estimation Method	Load Factor		Data Source
					Mean	Median	
Tractor	125-140	2006-2007	0.55	ECU	0.42	0.41	MSCD
Forklift	174	2005	0.3	ECU	0.15	0.1	MSCD
Dozer	75-450		0.59	ECU	0.39		Manufacturer
Excavator	95-500		0.57	ECU	0.43		Manufacturer

Based on electronic control unit (ECU) data, a correction factor was derived using the following formula:

$$CF = 1 - \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_{off} - X_i}{X_{off}}\right)$$

Where:

- CF = Correction factor, unitless
- X_{off} = Load factor in OFFROAD model
- X_i = Load factor provided by ECU

A correction factor of 0.67 was estimated and applied to the original OFFROAD load factors for airport ground support equipment (GSE), construction and mining, industrial, and oil drilling equipment. Other studies that evaluated the load factors used in OFFROAD came to similar conclusions (San Pedro Bay Ports, 2009). Table 7 lists the original OFFROAD and proposed load factors for each type of off-road rule equipment.

Table D-7: Original OFFROAD and New Load Factors (LF) by Equipment Type

Category	Equipment	OFFROAD Load Factor	Proposed Load Factor
Airport Ground Support Equipment (GSE)	A/C Tug Narrow Body	0.8	0.54
	A/C Tug Wide Body	0.8	0.54
	Baggage Tug	0.55	0.37
	Belt Loader	0.5	0.34
	Bobtail	0.55	0.37
	Cargo Loader	0.5	0.34
	Cargo Tractor	0.54	0.36
	Forklift (GSE)	0.3	0.20
	Lift	0.5	0.34
	Other GSE	0.5	0.34
	Passenger Stand	0.59	0.40
Construction and Mining	Bore/Drill Rigs	0.75	0.50
	Cranes	0.43	0.29
	Crawler Tractors	0.64	0.43
	Excavators	0.57	0.38
	Graders	0.61	0.41
	Off-Highway Tractors	0.65	0.44
	Off-Highway Trucks	0.57	0.38
	Other Construction Equipment	0.62	0.42
	Pavers	0.62	0.42
	Paving Equipment	0.53	0.36
	Rollers	0.56	0.38
	Rough Terrain Forklifts	0.6	0.40
	Rubber Tired Dozers	0.59	0.40
	Rubber Tired Loaders	0.54	0.36
	Scrapers	0.72	0.48
	Skid Steer Loaders	0.55	0.37
	Surfacing Equipment	0.45	0.30
	Tractors/Loaders/Backhoes	0.55	0.37
	Trenchers	0.75	0.50
Industrial	Aerial Lifts	0.46	0.31
	Forklifts	0.3	0.20
	Other General Industrial Equipment	0.51	0.34
	Other Material Handling Equipment	0.59	0.40
	Sweepers/Scrubbers	0.68	0.46
Oil Drilling	Drill Rig (Mobile)	0.75	0.50
	Workover Rig (Mobile)	0.75	0.50

3. Activity

The activity or annual operation of off-road equipment is measured in annual average hours of use and varies by equipment type and age. Staff derived activity values from the OFFROAD model and further adjusted the values based on Assembly Bill 8 2x reporting data.

a. OFFROAD Activity Data

The activity values in the OFFROAD model were derived from various survey data (ARB, 2007). Since these survey data were collected between 2003 and 2005, the effects of the recent economic recession (2007 – 2009) on activity are not included. In addition, much of the OFFROAD data are not specific to California fleets. To update the activity values and incorporate the effects of the economic recession, staff utilized new activity data (required by Assembly Bill 8 2x) to adjust the OFFROAD activity values.

b. Assembly Bill 8 2x Reporting Data

In 2009, the California legislature approved Assembly Bill 8 2x, which required ARB to amend certain sections of the off-road diesel regulation (AB 8 2x, 2009). As part of the amendments, ARB gave credit to fleet owners toward early requirements if they provided documentation that indicated reduced operation (i.e. hour meter readings, fuel purchase records, etc.) and/or reduced population of their off-road equipment since 2007. In order to receive credit for reduced usage, fleet owners had to provide equipment-specific activity values (hours/year) for the baseline year (2007) and reduced activity values for the same equipment in 2009. Fleet owners provided such data from late 2009 through April 2010 in accordance with requirements in the reduced activity reporting guide (ARB, 2009a). These data (ARB, 2009b) were used to adjust OFFROAD activity values to new baseline levels and to derive depressed activity values to reflect the recent economic recession.

To ensure that the activity data from AB 8 2x fleets were representative of the general California population, staff compared the fleet size (i.e. number of vehicles/fleet) of fleets with 2009 reduced activity data (96 fleets) to all 828 large fleets in DOORS. Figure 1 compares the two data sources by fleet size. Although the fleet sizes were not identical for the sample and DOORS fleets, the average fleet size and distribution in the two data sources were not significantly different (95% confidence level). In addition, the sample fleets were similar in terms of total horsepower for specific equipment types when compared to all DOORS large fleets. As a result, staff determined that data from the AB 8 2x fleets were a suitable surrogate for all California fleets.

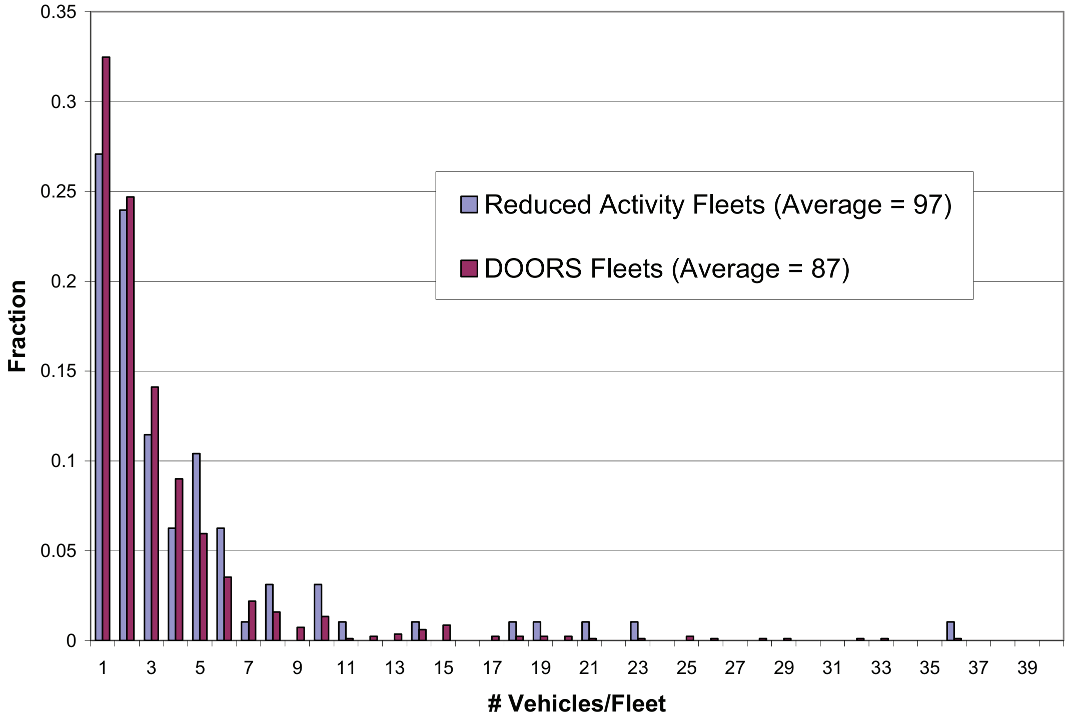


Figure D-1: Fleet Population Comparisons

c. Calculation of New Activity Values

Staff calculated new activity values assuming that the AB8 2x reporting data for 2007 activity represented baseline activity under normal economic conditions. This assumption was based on the fact that 2007 represents an approximate midpoint between peak economic activity in 2005/2006 and the recessed economy of 2009.

For each equipment type where activity data were available, staff calculated an age-specific ratio of the 2007 activity to the OFFROAD activity. Staff then calculated an overall weighted ratio to use as an adjustment factor to OFFROAD activity values. This weighted ratio was applied to age-specific activity values in OFFROAD for each equipment type to derive new baseline activity values. Staff also performed an identical analysis using the AB8 2x 2009 reduced activity data whereby a weighted ratio of the 2009 activity relative to the OFFROAD activity was calculated and used to derive the depressed activity values for 2009/2010. For equipment types with insufficient activity data, the average value of the ratios in the general equipment category to which the equipment belonged was used as the adjustment factor. For example, since sufficient activity data for bore/drill rigs were unavailable, the average of the adjustment factors for those pieces of equipment in the construction and mining sector with sufficient data were used to derive activity values for bore/drill rigs. Table 8 provides the adjustment factors used to derive new baseline and reduced activity values from OFFROAD values.

Table D-8: Activity Adjustment Factors Applied to OFFROAD Values

Equipment	2007 Baseline Activity Weighted Average Correction Factor	2009 Reduced Activity Weighted Average Correction Factor
Scrapers	0.52	0.22
Tractors/Loaders/Backhoes	0.62	0.40
Rubber Tired Loaders	1.03	0.69
Excavators	0.44	0.27
Crawler Tractors	0.45	0.29
Rollers	0.48	0.34
Forklifts	0.45	0.26
Graders	0.71	0.50
Other Construction Equipment	0.70	0.46
Off-Highway Trucks	0.74	0.46
Rough Terrain Forklifts	0.21	0.22
Aerial Lifts	0.72	0.65
Workover Rig (Mobile)	0.40	0.24
Skid Steer Loaders	0.35	0.33
Cranes	0.40	0.25
Pavers	0.42	0.34
All Other Airport GSE	0.54	0.37
All Other Construction Equipment	0.54	0.37
All Other Industrial Equipment	0.58	0.45
All Other Oil Drilling Equipment	0.40	0.24

After calculating the age-specific activity values, staff determined the fraction of vehicles in California experiencing reduced activity. Specifically, staff determined what proportion of the overall reductions in horsepower hours in 2009 were met solely by reduced activity since there was a portion of fleets that had experienced reduced activity due to reduced population (i.e. horsepower) rather than usage (i.e. hours/year). Based on an analysis of AB 8 2x reporting data (ARB, 2009b), staff determined that approximately 34% of the reductions in horsepower hours could be attributed to reduced activity. The other reductions in horsepower hours were met by reductions in vehicle population. As a result, the remaining 66% of the vehicles were assumed to operate at the 2007 baseline activity values. Staff used these two values as weighting factors to derive the new adjusted activity values to be used in the emissions inventory. Table 9 provides a comparison of the average activity by equipment type in the original OFFROAD model and the modified values in the new emissions inventory model.

Table D-9: OFFROAD and Modified Average Activity Values (hours/year)

Category	Equipment	OFFROAD Activity	Modified Activity
Airport Ground Support Equipment (GSE)	A/C Tug Narrow Body	625	301
	A/C Tug Wide Body	759	365
	Baggage Tug	1,392	669
	Belt Loader	974	468
	Bobtail	683	328
	Cargo Loader	906	436
	Cargo Tractor	1,309	630
	Forklift (GSE)	743	357
	Lift	791	380
	Other GSE	922	443
	Passenger Stand	70	34
Construction and Mining	Bore/Drill Rigs	811	392
	Cranes	1,252	444
	Crawler Tractors	1,013	409
	Excavators	1,396	546
	Graders	929	610
	Off-Highway Tractors	1,091	534
	Off-Highway Trucks	1,958	1,293
	Other Construction Equipment	690	429
	Pavers	821	324
	Paving Equipment	829	400
	Rollers	695	299
	Rough Terrain Forklifts	1,123	237
	Rubber Tired Dozers	1,589	767
	Rubber Tired Loaders	957	893
	Scrapers	1,092	453
	Skid Steer Loaders	834	297
	Surfacing Equipment	446	215
	Tractors/Loaders/Backhoes	942	512
	Trenchers	618	298
Industrial	Aerial Lifts	384	266
	Forklifts	1,800	690
	Other General Industrial Equipment	1,425	766
	Other Material Handling Equipment	1,318	709
	Sweepers/Scrubbers	1,220	656
Oil Drilling	Drill Rig (Mobile)	3,000	1,025
	Workover Rig (Mobile)	3,000	1,025

Figures 2 – 5 provide examples of the various age-specific activity values used in the OFFROAD model and the proposed values for scrapers, tractors/loaders/backhoes, excavators, and crawler tractors. These equipment types account for over 40 percent of the total horsepower in the California off-road diesel equipment population. As seen from the figures, the original OFFROAD activity values are much higher than the proposed baseline activity values in 2007. Also shown is the effect of the economic recession on 2009 activity values. For designated low-use vehicles of all equipment types, staff assigned a value of 150 hours/year, which is the threshold for receiving a low-use designation. Staff also examined activity data from a 2003 TIAX report (TIAX, 2003) a 2007 Eastern Research Group (ERG) study (ARB,2008). These data agreed reasonably well with the baseline activity values derived for 2007.

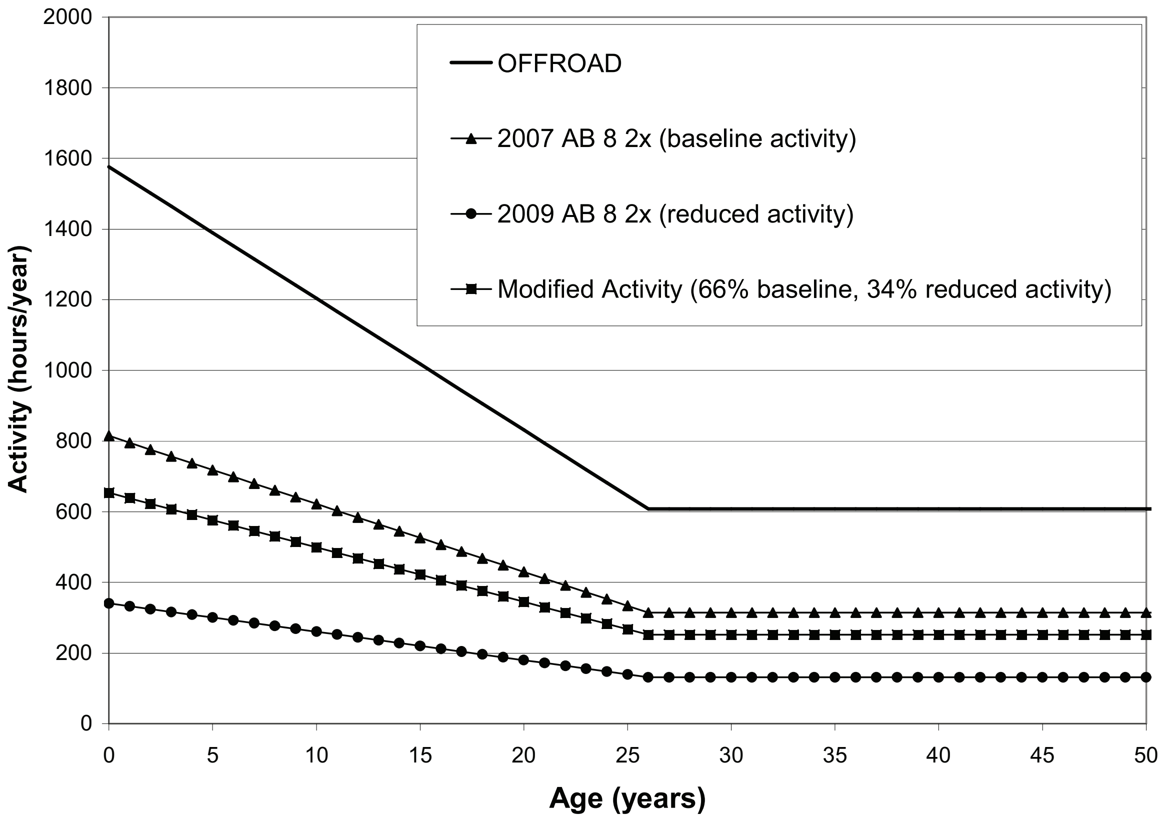


Figure D-2: Scrapers Activity Values

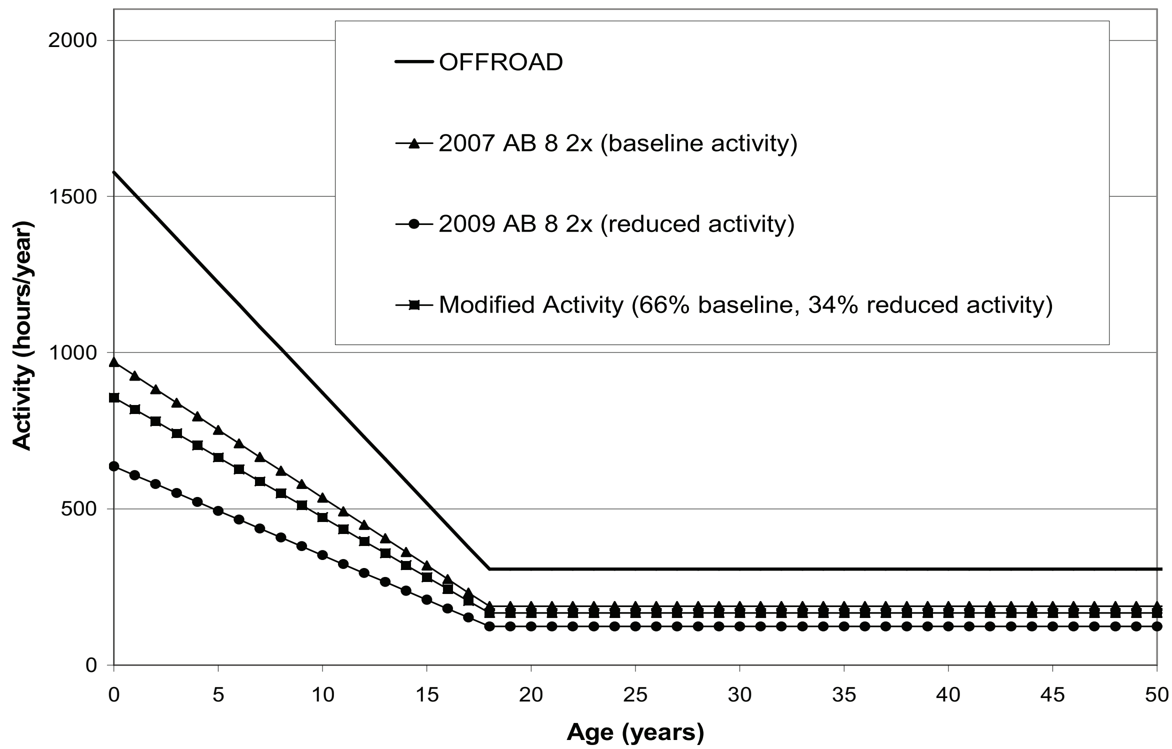


Figure D-3: Tractors/Loaders/Backhoes Activity Values

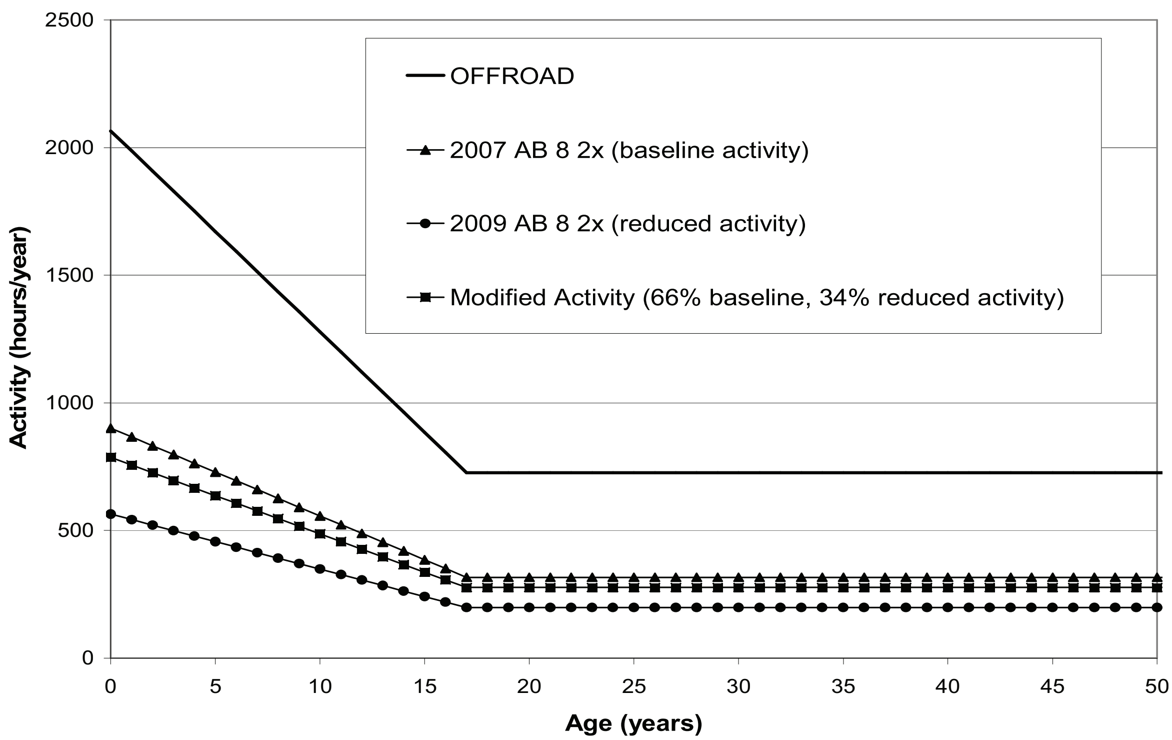


Figure D-4: Excavators Activity Values

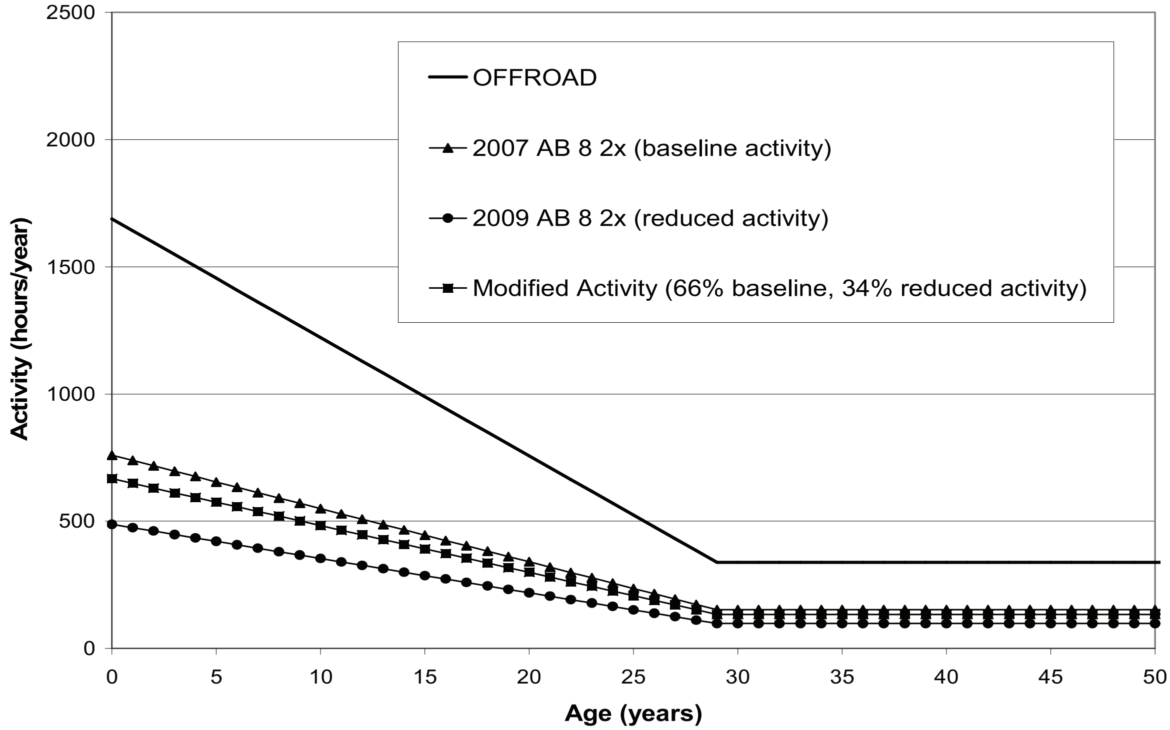


Figure D-5: Crawler Tractors Activity Values

4. Age Distributions and Economic Recovery Factors

a. Background

The age distribution of off-road diesel equipment has important implications for the emissions inventory. In general, an older vehicle will produce more emissions than a newer vehicle operating under the same conditions. In light of the economic recession, it is important to assess the impacts of the economy on sales of new off-road diesel equipment. Depending on the state of the economy in a given calendar year, one of several scenarios will occur regarding the sales of the new equipment. These scenarios will impact the relative proportion of the new model year equipment in the age distribution. These scenarios include:

- **New equipment sales are higher than expected equipment sales**. The proportion of the new model year equipment pieces in the age distribution will therefore be *higher* than the proportion in a baseline (no impact) age distribution.
- **New equipment sales are lower than expected equipment sales**. The proportion of the new model year equipment pieces in the age distribution will therefore be *lower* than the proportion in a baseline (no impact) age distribution.
- **New equipment sales are equal to expected equipment sales**. The proportion of the new model year equipment pieces will therefore be *equal* to the proportion in a baseline (no impact) age distribution age distribution.

The original OFFROAD emissions inventory did not consider these effects on the age distribution. Instead, staff derived a base year age distribution and the equipment was assumed to follow a specific attrition rate, which was used to derive future age distributions. To account for the impacts of the California economy on future age distributions, staff estimated the impact of new off-road diesel equipment sales on the age distributions. The 2009 age distributions and average ages of off-road diesel equipment were based on data from the Diesel Off-road On-line Reporting System (DOORS) database (DOORS, 2010). Once future age distributions were derived, they were used to calculate average ages by calendar year, which in turn were used to derive economic recovery factors as used in the Off-road Simulation Model (OSM) to model fleet turnover (described in Attachment A).

b. Methodology and Assumptions

Staff estimated future age distributions based on the assumption that trends in sector-specific gross domestic product (GDP) directly impact the relative proportion of a given model year equipment in a baseline age distribution. The baseline age distribution is the age distribution that is expected without the impacts of GDP on sales patterns. So, during a calendar year with high economic growth (i.e. high GDP), the proportion of the corresponding model year equipment within the age distribution is increased relative to the baseline age distribution. Conversely, during times of economic recession, less new equipment is purchased and the proportion of the corresponding model year equipment within the age distribution is decreased relative to the baseline age distribution. Staff considered the effects of sales patterns within the following individual sectors of off-road equipment: construction and mining, industrial, airport ground support equipment (GSE), and oil drilling. Each of these sectors is described below.

c. Construction and Mining Equipment

<u>Correlation between new equipment sales and California GDP</u>

Prior to estimating future age distributions, it was first necessary to validate the assumption that new off-road diesel equipment sales were correlated to California specific GDP sectors. In order to do this, staff examined in detail the relationship between financed sales of new off-road diesel construction equipment in California and California construction GDP. California equipment sales for new construction equipment were obtained from Equipment Data Associates' database of Uniform Commercial Code 1 (UCC1) filings (EDA, 2010). UCC1 forms are financing statements supplied by lenders to the California Secretary of State to indicate a security interest in financed property. For this analysis, staff analyzed UCC1 forms concerning financed new sales of construction equipment in California from 1990 to 2009. Although the UCC1 filings did not capture all data regarding the off-road diesel equipment sales in California, staff assumed the relative amount of financed new equipment sales (i.e. ratio of sales from one year to the next) would be sufficient to model the trend in sales of new equipment. GDP data were obtained from the United States Bureau of Economic

Analysis (BEA, 2010). Using multiple linear regression with time and construction GDP as predictor variables, staff found a strong correlation ($R^2 = 0.93$) between financed new construction equipment sales in Year *y* and the GDP from Year *y*+1, the time lag indicating that the construction industry purchasing behavior was impacted before a decline in economic activity. Figure 6 provides a comparison of the actual sales of new construction equipment and the predicted sales for calendar years 1990-2009. The average difference between the actual sales number and the predicted sales number was 3.6%. Based on these results, staff determined that construction GDP is a useful predictor of new construction equipment sales.

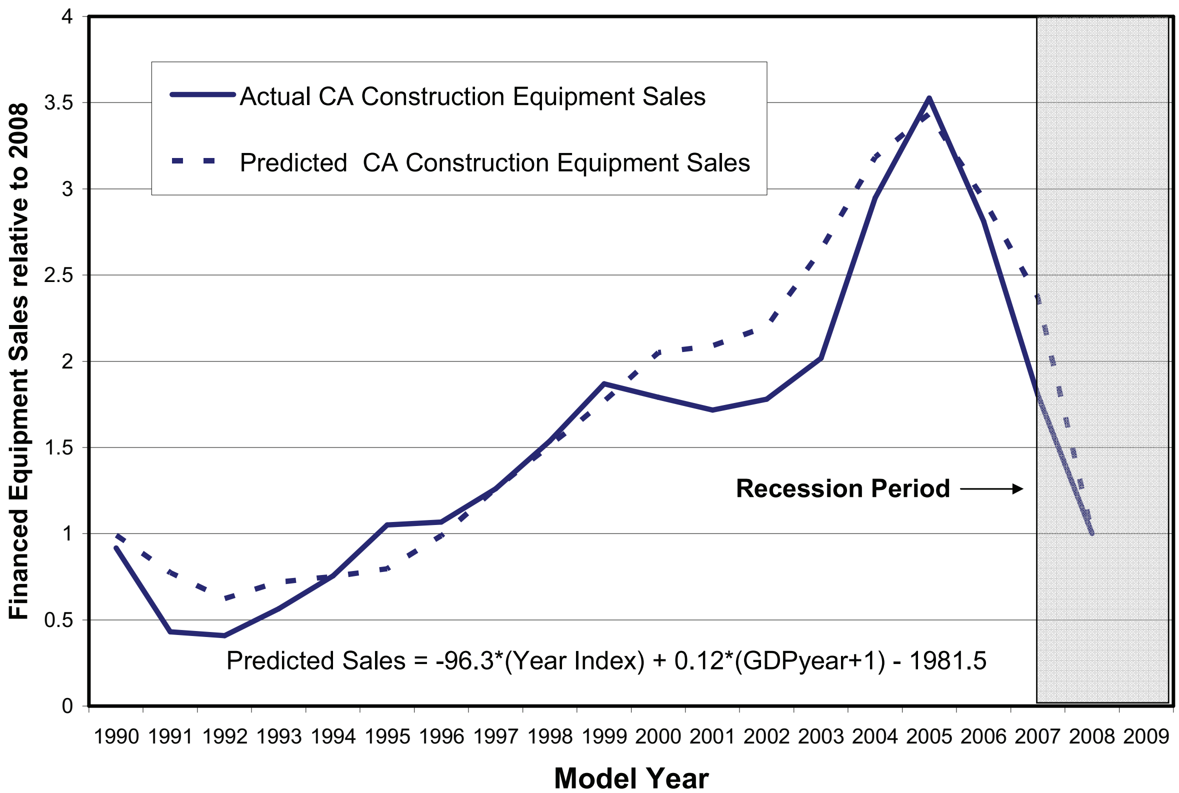


Figure D-6: Actual and Predicted New Construction Equipment Sales

<u>Derivation of Model Year Adjustment Factors</u>

The correlation between construction GDP and construction sales enabled staff to predict historical construction equipment sales prior to 1990 for which no sales data was available. Historical GDP values (< 1990) were provided by the U.S. Bureau of Economic Analysis (BEA, 2010). To incorporate the impact of construction equipment sales on the age distribution, staff first calculated the ratio between sales of construction equipment and the expected number of sales if no factor (i.e. construction GDP) had caused variation for each particular model year. These ratios served as model year adjustment factors for future calendar years. The actual/predicted sales as well as the best-fit line (i.e. expected equipment sales) associated with "no GDP impact" are shown in Figure 7 below.

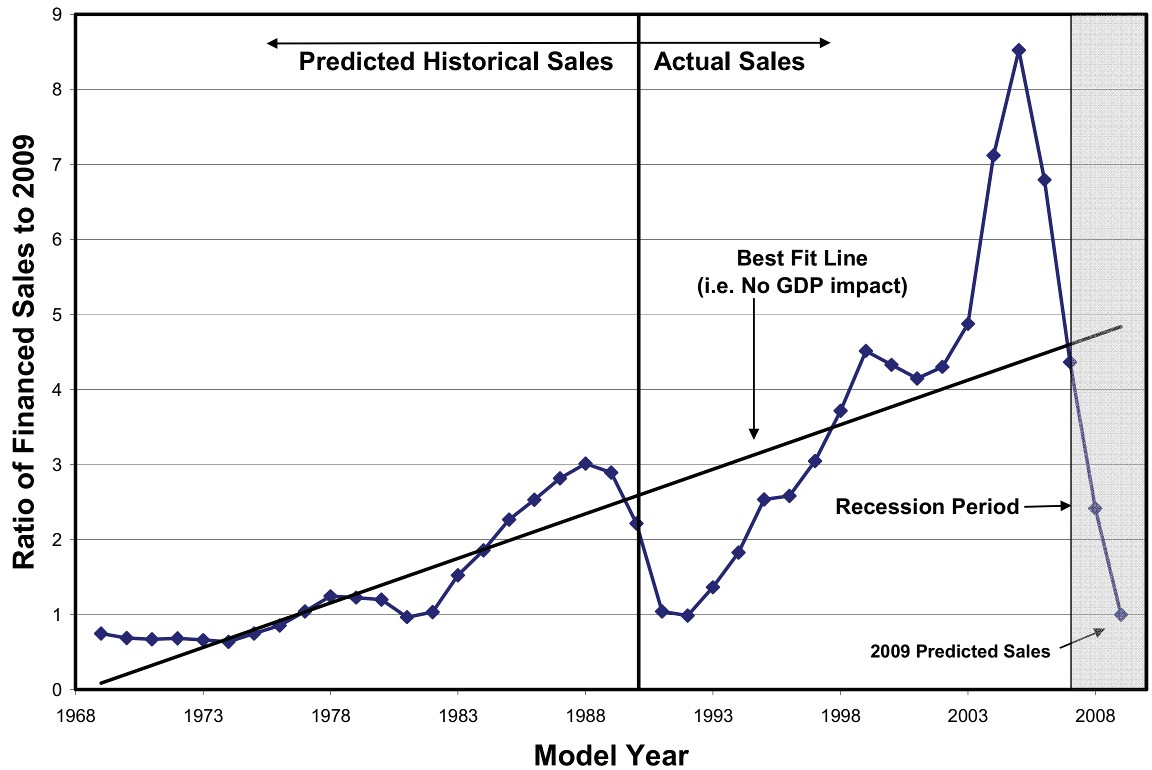


Figure D-7: California New Construction Equipment Sales with Best Fit Line

<u>Application of Model Year Adjustment Factors and Derivation of Future Calendar Year Age Distributions</u>

The construction and mining equipment age distribution for 2009 was derived from DOORS. Staff derived the baseline age distribution (i.e. no GDP impact) by dividing the 2009 DOORS age distribution by the appropriate model year adjustment factors to remove the effects of the economy. An example of the 2009 age distribution with the adjusted baseline age distribution is shown in Figure 8 for the construction and mining category. Although the 2009 age distribution from DOORS was used only for the calendar year 2009 age distribution, the derived baseline age distribution was used as a starting point to predict future age distributions.

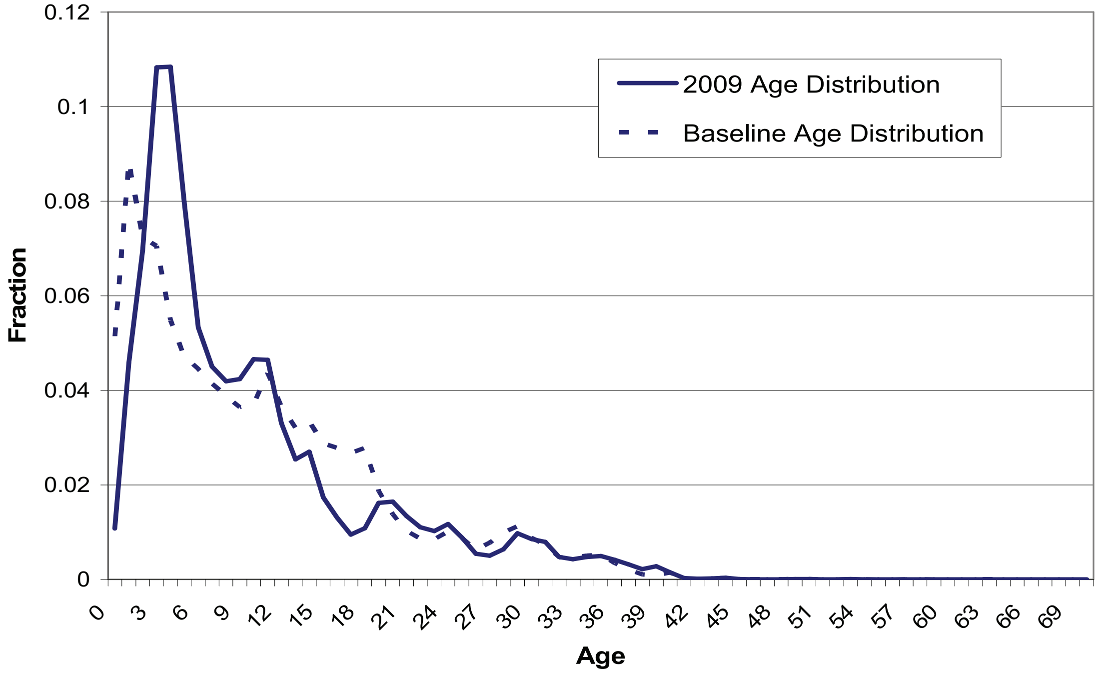


Figure D-8: 2009 and Baseline Age Distributions for Construction and Mining Equipment

Staff derived the model year proportions in future age distributions (2010 – 2030) by multiplying the fraction of each model year in the baseline age distribution by the appropriate model year adjustment factor. By doing this, the effects of the economy were implicitly incorporated into the age distribution. The model year adjustment factors for model years 2010 and greater had to be derived based on assumptions about the recovery in new equipment sales. In order to derive these adjustment factors, staff looked at a few economic recovery scenarios for sales. The fast scenario assumed construction equipment sales return to the average historical trend in 2017. The slow growth scenario assumed growth at the previous historical rate but never returns to baseline conditions. The average growth scenario represented an average of the fast and slow growth scenarios. For the purposes of the emissions inventory, the age distributions derived under the average growth scenario were used up until 2017. After 2017, new equipment sales were assumed to continue at the same rate as the average scenario until they reached the expected sales ratio in 2025. This recovery scenario for the construction sector is shown in Figure 9. Similar to the pre-2010 model year adjustment factors, the ratio of new equipment sales to the expected equipment sales (best fit line) was used to calculate a model year adjustment factor. An example of the age distribution for construction and mining equipment in 2009, 2017 and 2030 is shown in Figure 10. As expected, the age distribution begins to resemble the baseline age distribution (Figure 8) as time goes on and the economy recovers. It should be noted that these age distributions were not directly used to predict age distributions in the future, which is done by the Off-road Simulation Model (OSM). Instead they were used as the basis for calculating economic recovery factors as described below.

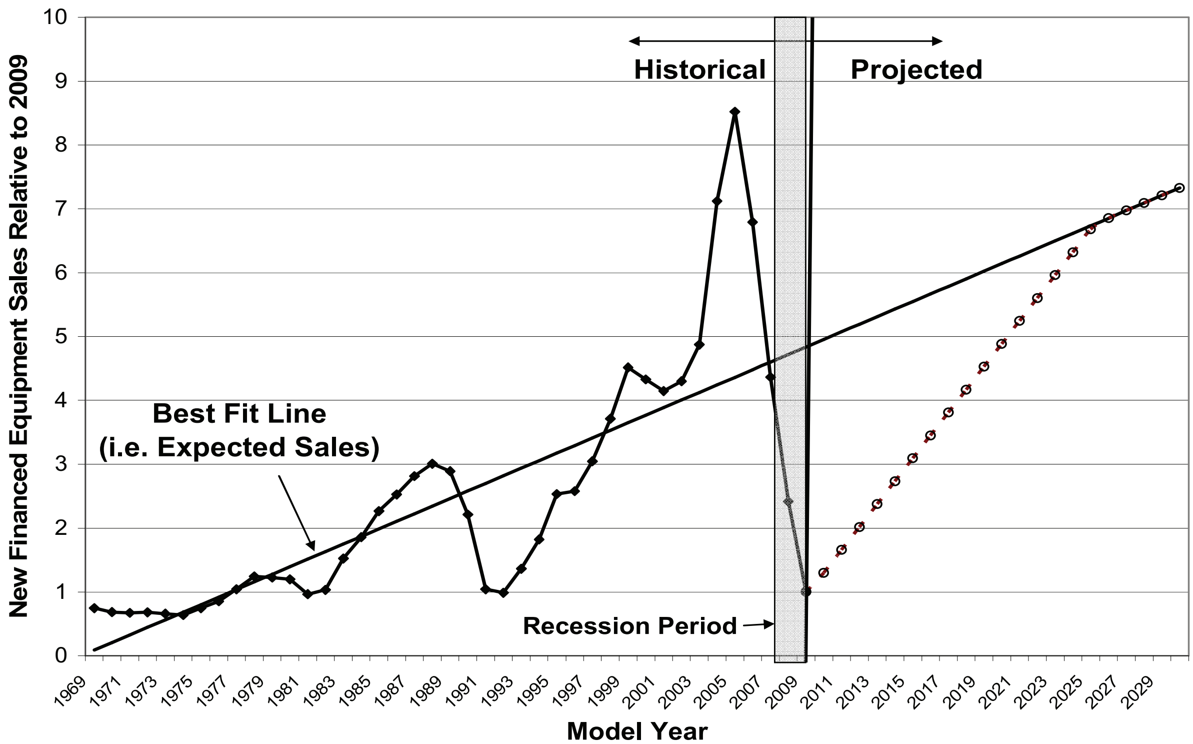


Figure D-9: Recovery Scenario Projections for New Construction Equipment Sales

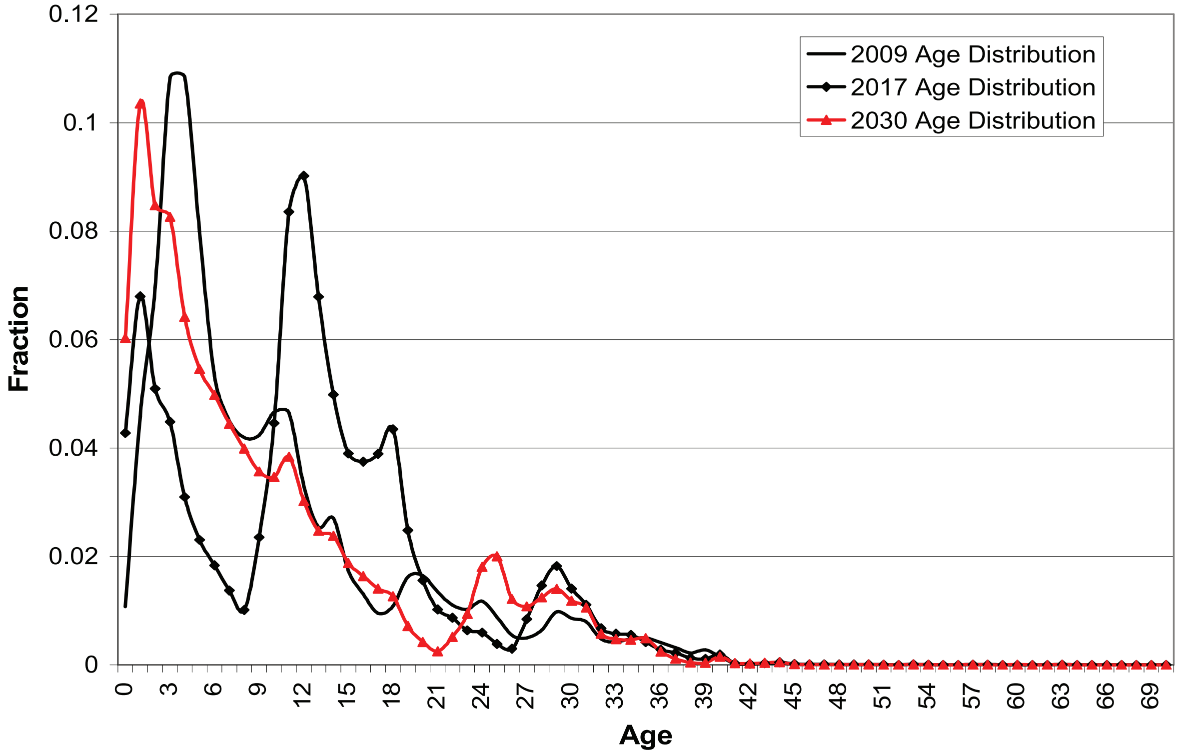


Figure D-10: Age Distribution of Construction and Mining Equipment in Average Recovery Scenario

<u>Calculation of Average Ages and Economic Recovery Factors</u>

Staff calculated the weighted average age value for the construction and mining sector age distributions for 2009 to 2030 (Figure 11). As seen from the figure, the average age increases during the initial years after 2009. The ratios in the figure were used as economic recovery factors in the Off-road Simulation Model (described in Attachment A). The initial increase in the average age is a result of much fewer new vehicles entering the fleet relative to the historical average. As the economy recovers, the average age begins to decrease as older equipment is turned over and newer equipment, at rates close to the historical average, enters the fleet.

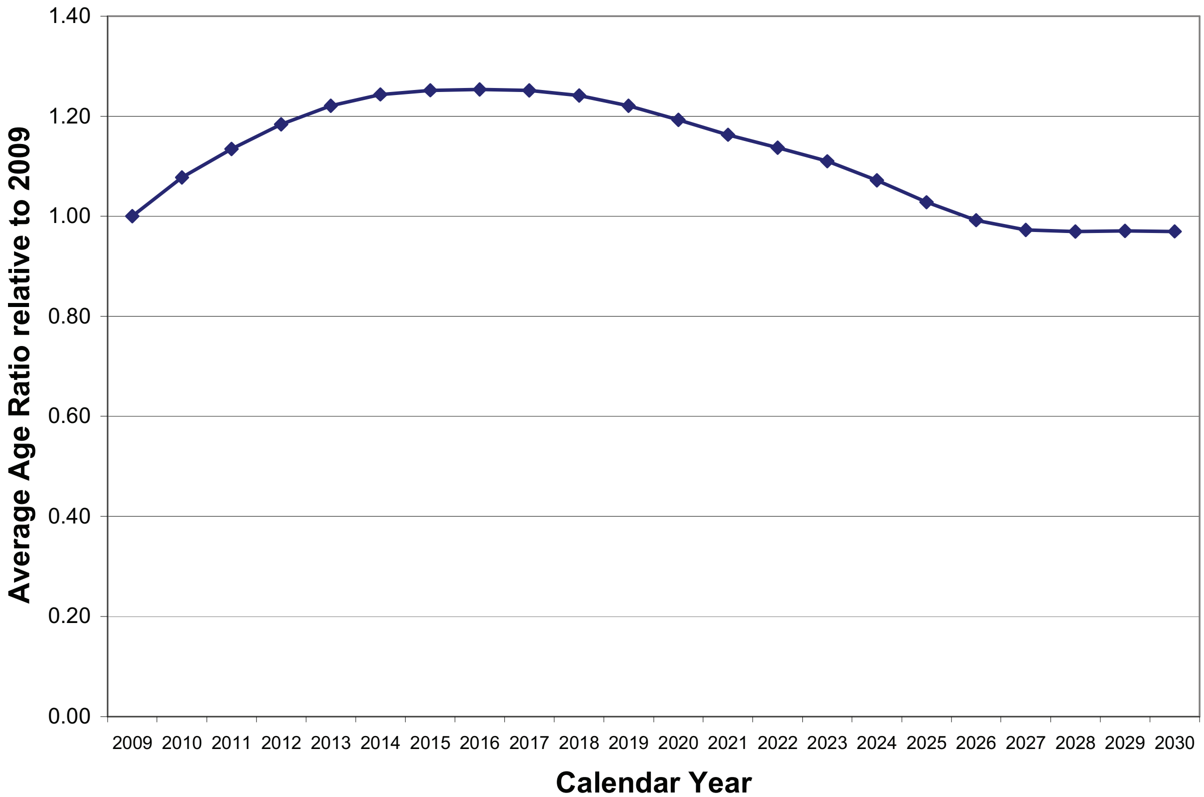


Figure D-11: Average Ages of Construction Equipment (2009 – 2030)

d. Industrial Equipment

Since most of the industrial companies in the DOORS were suppliers of construction equipment and construction building materials, staff assumed that sales patterns of new industrial equipment followed the same pattern in sales of new construction equipment. As a result, the same model year adjustment factors were applied to industrial equipment as were applied to construction equipment. The approach for deriving future age distributions was the same as construction equipment. The average ages of industrial equipment are shown in Figure 12.

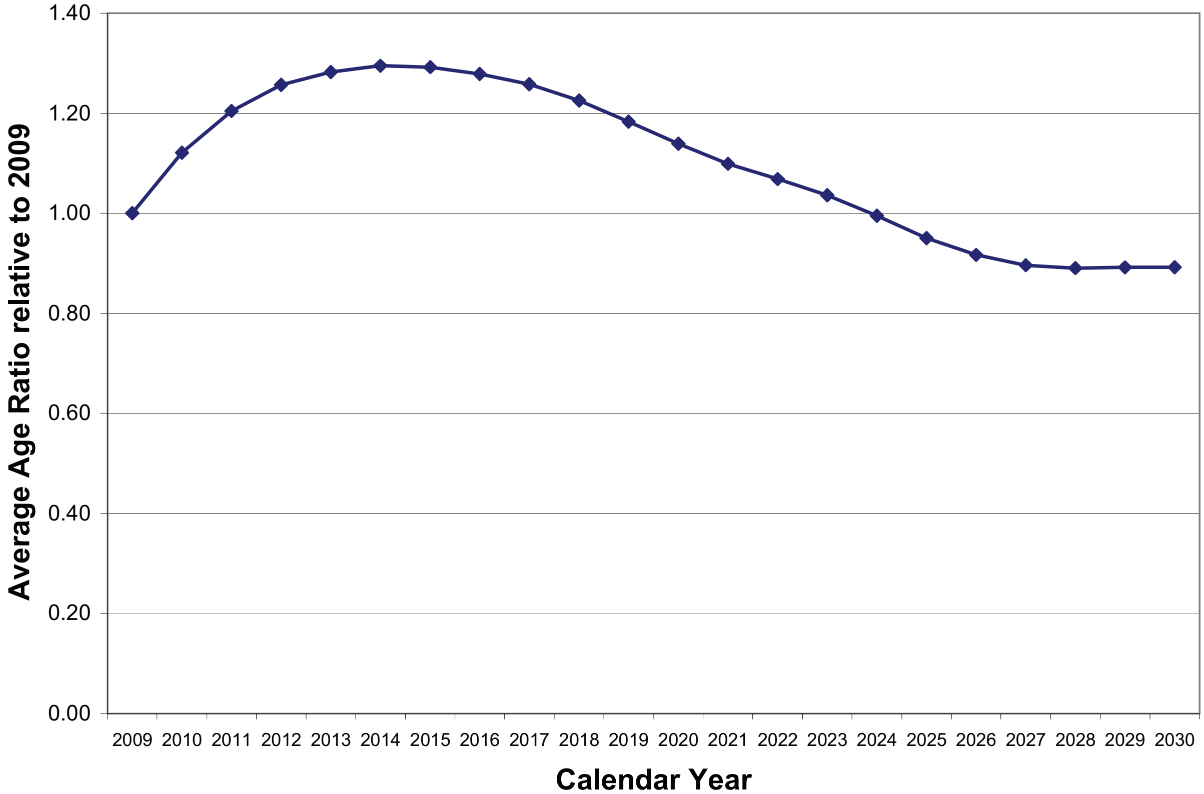


Figure D-12: Average Ages of Industrial Equipment (2009 – 2030)

e. Airport Ground Support Equipment (GSE)

Based on the strong correlation between financed new construction equipment sales and construction GDP, staff made the assumption that sales of new GSE equipment were also strongly correlated to GDP. As a result, a similar approach was taken for these equipment types using California air transport GDP as a surrogate for sales of GSE. The determination of model year adjustment factors, future age distributions and average ages was determined in the same manner as the construction and industrial equipment categories. The age distribution derived under the average recovery scenario was used for calculations in the emissions inventory. The average recovery scenario for GSE is shown in Figure 13. The average ages of GSE under the average recovery scenario are shown in Figure 14.

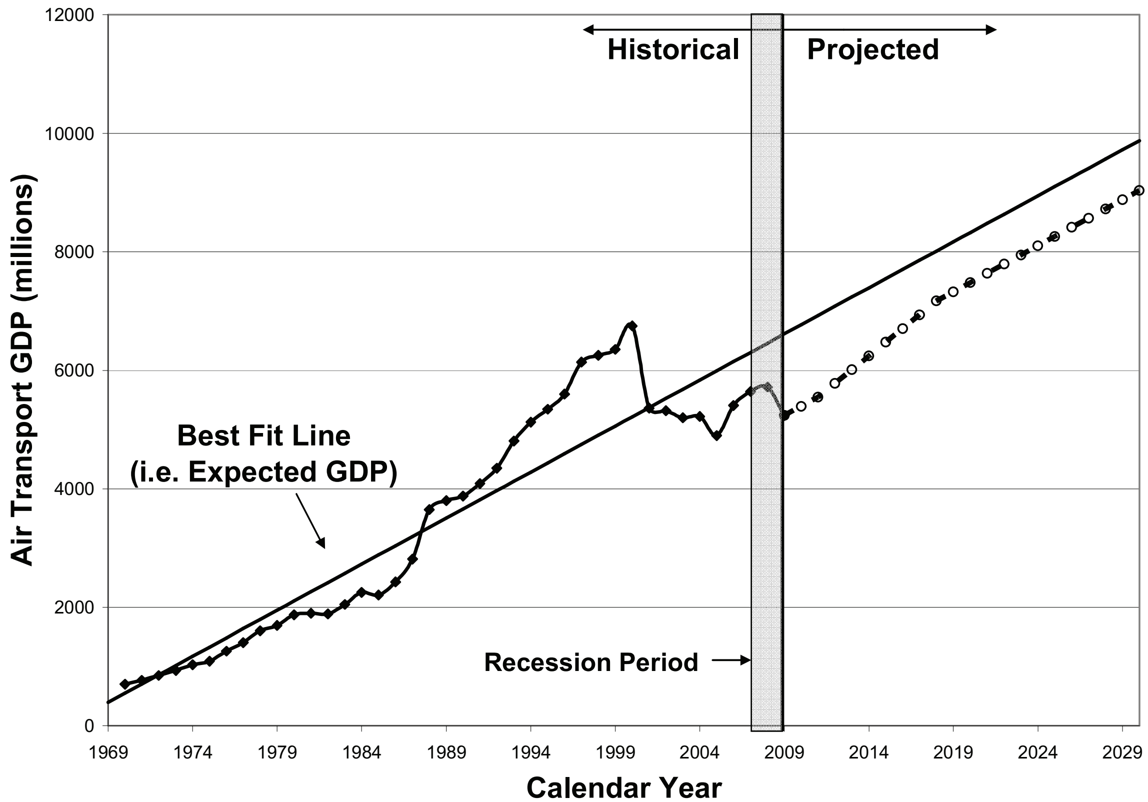


Figure D-13: Recovery Scenario Projections for Air Transport GDP

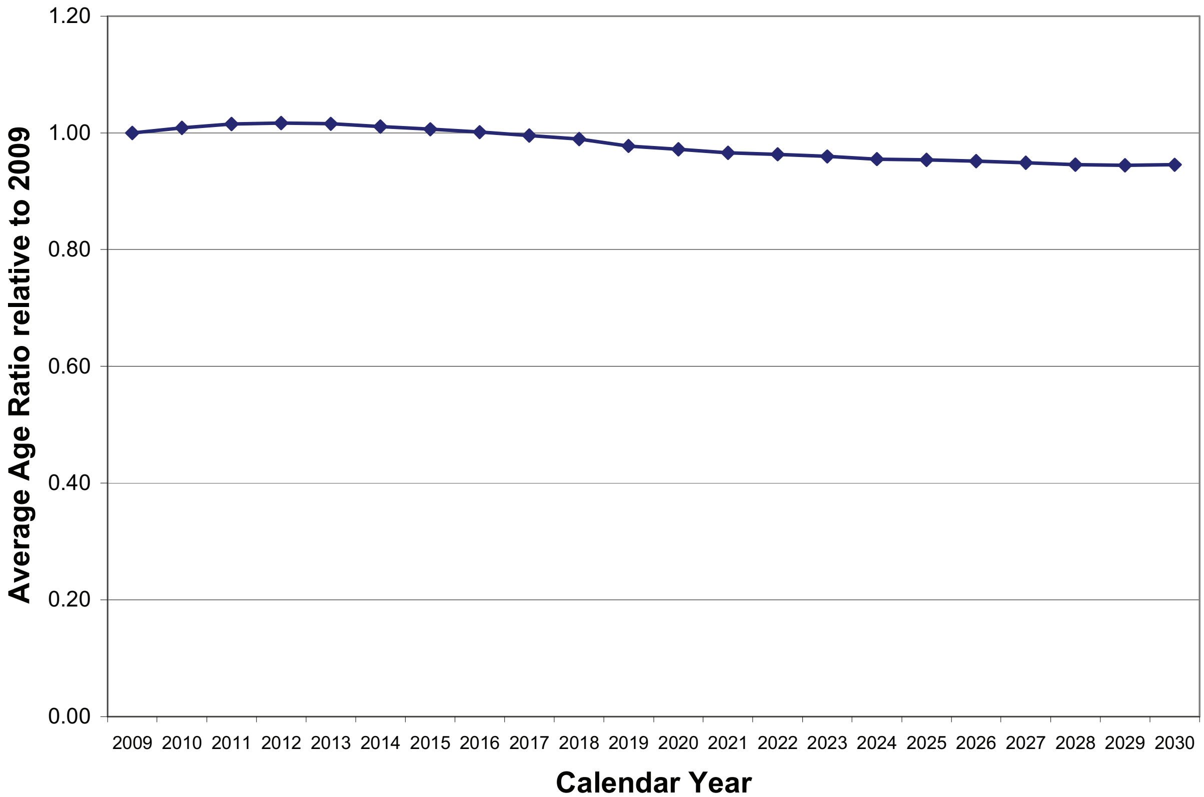


Figure D-14: Average Ages of Airport Ground Support Equipment (2009 – 2030)

f. Oil Drilling Equipment

Due to the strong historical variability in oil drilling GDP, it was difficult to establish a strong correlation between sales of oil drilling and oil drilling GDP. As a result, the age distributions and average ages for all future years were assumed to be equivalent to the 2009 DOORS age distribution.

5. Emission Factors

Emission factors are composed of zero-hour emission rates[1] and deterioration rates. The emission factors can be expressed by the following equations.

For cumulative hours (CHrs) less than or equal to 12,000 hours:

---

[1] Section I-G of ARB's Mail-Out#: MSC 99-32 contains a more detailed discussion of how each emission rate is derived for Tier 0 or uncontrolled, through Tier 3. The Tier 4 emission rates were adopted December 7, 2005, as detailed in "Amendments to the California Exhaust Emission Standards and Test Procedures for New 2008 and Later Tier 4 Off-Road Compression-Ignition Engines and Equipment, Part 1-C" (http://www.arb.ca.gov/regact/offrdcie/offrdcie.htm)

$$EF = Zh + dr * CHrs$$

For cumulative hours (CHrs) greater than 12,000 hours:

$$EF = Zh + dr * 12,000$$

Where:

EF	= Emission factor (g/bhp-hr)
Zh	= Zero-hour emission rate when the equipment is new (g/bhp-hr)
Dr	= Deterioration rate or the increase in zero-hour emissions as the equipment is used (g/bhp-hr^2)
CHrs	= Cumulative hours or total number of hours accumulated on the equipment; maximum value is equal to 12,000 hours

The diesel emission factors in the model are in grams per brake horsepower-hour and vary by fuel type, horsepower, and model year as shown in Attachment D. To estimate fuel consumption, an emission factor is replaced with a brake-specific fuel consumption (BSFC) value (lb/hp-hr). BSFC values are used from the U.S. EPA NONROAD model (EPA, 2004b). These BSFC values are lower than the values used in the original rulemaking and are more consistent with input ARB staff received from stakeholders.

6. Emission Factor Deterioration

Deterioration rates[2] are in units of grams per brake horsepower-hour^2 and are defined as the change in emissions as a function of usage. These are based on the deterioration rates of on-highway diesel-powered engines with similar horsepower ratings. The rate of emissions changes over time as a result of wear on various parts of an engine due to use. Originally, the deterioration rate continued to increase throughout the entire useful life of the equipment. Based on discussions with stakeholders and industry, staff concluded that the deterioration for off-road diesel equipment was over-estimated because diesel engines are rebuilt after approximately 12,000 hours of use (ARB, 2007). It is assumed that an engine would be rebuilt back to the standard of that particular emissions tier (varies by model year of the engine). Taking engine rebuilds into account, staff has updated the diesel emission factor deterioration to cap at 12,000 hours. As a result, once an engine's cumulative hours equals 12,000 hours, the deteriorated emission factor is assumed to be constant at that rate for the rest of the life of the equipment.

---

[2] Section I-I of ARB's Mail-Out#: MSC 99-32 contains a more detailed discussion of how each deterioration rate is derived for uncontrolled, and Tier 1 to Tier 3. The Tier 4 deterioration rates were derived using the ratio to the standard of Tier 3.

7. Fuel Correction Factors

The fuel correction factors (FCF) used in the emissions inventory model are dimensionless multipliers applied to the basic exhaust emission rates that account for differences in the properties of certification fuels compared to those of commercially dispensed fuels. In instances where engines or vehicles are not required to certify, the FCFs reflect the impact in changes of dispensed fuel over time as refiners respond to changes in fuel specific regulations compared to the fuel used to obtain the test data. The FCFs used in the model were specific to horsepower group and model year and were based on data described in a 2005 OFFROAD Modeling Change Technical Memo (ARB, 2005).

D. Spatial Allocation and Emissions Projections

1. Spatial Allocation

The spatial allocation distributes the entire population of off-road rule equipment to separate geographic regions within the State. Since the precise location of every piece of mobile off-road equipment is unknown, staff allocated equipment to the 15 air basins within California based on equipment population surrogates. These surrogates are indicators of the population of equipment operating within each air basin. Below, the spatial allocations for the four categories of off-road rule equipment are described.

a. Airport Ground Support Equipment

The allocation of airport ground support equipment (GSE) is based on airport-specific data obtained from the Department of Transportation's (DOT) 1997 Flight Ops data (ARB, 2007). The allocation was reviewed by the GSE's industry representative/consultant as part of the original rulemaking process in 2005. This data was also utilized as the basis of allocating equipment used in GSE applications in the OFFROAD model. Table 10 contains the statewide allocation for airport ground support equipment.

Table D-10: Airport Ground Support Equipment Statewide Allocation by Air Basin

Air Basin	Abbreviation	Statewide Allocation
Great Basin Valleys	GBV	0.0%
Lake County	LC	0.0%
Lake Tahoe	LT	0.0%
Mountain Counties	MC	0.0%
Mojave Desert	MD	0.1%
North Coast	NC	0.2%
North Central Coast	NCC	0.4%
Northeast Plateau	NEP	0.0%
South Coast	SC	50.2%
South Central Coast	SCC	0.6%
San Diego	SD	7.7%
San Francisco	SF	36.0%
San Joaquin Valley	SJV	1.0%
Salton Sea	SS	0.5%
Sacramento Valley	SV	3.2%
Total		**100%**

b. Construction and Mining Equipment

For construction fleets, staff assumed that the regional allocation is proportional to the regional rate of population growth. The Demographic Research Unit of California Department of Finance (DOF, 2009) provides population estimates and projections by county. The populations within counties were then distributed into air basins based on census data. Staff calculated the year-to-year increase in population within each air basin and used the fraction of the population growth in air basin versus statewide growth as a surrogate for construction fleet regional allocation. The resulting allocations are shown in Table 11 for 2009, 2014 and 2020. For years between DOF projections, staff interpolated regional allocation factors linearly.

Table D-11: Construction and Mining Equipment Statewide Allocation by Air Basin

Air Basin	OFFROAD Allocation	Updated 2009 Allocation	Updated 2014 Allocation	Updated 2020 Allocation
GBV	0.3%	0.01%	0.1%	0.1%
LC	0.5%	0.1%	0.2%	0.2%
LT	0.6%	0.2%	0.2%	0.2%
MC	3.4%	0.4%	0.9%	1.5%
MD	2.5%	2.5%	2.9%	3.4%
NC	2.0%	0.4%	0.5%	0.7%
NCC	3.0%	1.7%	1.6%	1.6%
NEP	0.3%	0.0%	0.1%	0.3%
SC	36.0%	35.6%	36.6%	37.7%
SCC	6.8%	3.0%	3.0%	3.0%
SD	9.1%	10.3%	8.6%	6.5%
SF	14.7%	21.0%	16.0%	10.1%
SJV	9.8%	13.8%	17.4%	21.7%
SS	2.1%	3.1%	3.4%	3.8%
SV	9.1%	7.8%	8.5%	9.3%
Total	**100%**	**100%**	**100%**	**100%**

c. Industrial Equipment

Since industrial operations are primarily situated in non-residential and less populated areas typically zoned for commercial or industrial use, human population is not the best surrogate on which to base the allocation of industrial equipment. County-level employment data was obtained from the Employment Development Department (EDD) for 2000 to 2008 based on the annual compilation of the Quarterly Census of

Employment & Wages (ES202) for the following Divisions: (01) Goods- Producing and (31) Manufacturing (EDD, 2010).

The county-specific data was distributed into air basins using data obtained from the 2000 U.S. Census overlaid with available air basin/county geographic information system (GIS) map layers. Actual EDD employment data were used from 2000 to 2008 to determine the county fraction for allocation of industrial equipment. Since the EDD does not provide any future forecasts for 2009 and beyond, the allocation of industrial equipment is based on data for 2008.

Table 12 below compares the updated allocation of industrial equipment by air basin with the original values from OFFROAD.

Table D-12: Industrial Equipment Statewide Allocation by Air Basin

Air Basin	OFFROAD Allocation	Updated Allocation
GBV	0.0%	0.0%
LC	0.0%	0.0%
LT	0.1%	0.1%
MC	0.4%	0.5%
MD	0.6%	2.2%
NC	0.6%	0.6%
NCC	1.1%	1.5%
NEP	0.1%	0.1%
SC	55.7%	46.6%
SCC	2.9%	3.9%
SD	6.2%	6.9%
SF	23.9%	23.9%
SJV	5.0%	9.2%
SS	0.4%	1.0%
SV	2.9%	3.5%
Total	**100%**	**100%**

d. Oil Drilling Equipment

County-level registration data of off-road equipment, used specifically for oil drilling applications, was obtained from the ARB's Portable Equipment Registration Program (PERP) Database in 2004 (ARB,2007), and was used to allocate oil drilling equipment in the OFFROAD model. This allocation was reviewed by industry and key stakeholders as part of the original rulemaking process in 2005. Table 13 contains the statewide allocation for oil drilling equipment. The allocation of oil drilling equipment in Table 13 is in good agreement with the number and location of rotary rigs operating within California (Baker Hughes, 2010).

Table D-13: Oil Drilling Equipment Statewide Allocation by Air Basin

Air Basin	Statewide Allocation
GBV	0.0%
LC	0.0%
LT	0.0%
MC	0.0%
MD	0.0%
NC	0.0%
NCC	0.0%
NEP	0.0%
SC	13.0%
SCC	8.5%
SD	0.0%
SF	2.7%
SJV	61.1%
SS	0.0%
SV	14.5%
Total	**100%**

2. Emissions Projections

The emissions inventory for off-road equipment is a function of engine emission factors and total equipment activity. Total equipment activity describes both the population of vehicles and much how the vehicles are operated. In order to project future off-road vehicle emissions, staff must estimate how these variables change over time. Emission factors change as a result of new engine technologies that are adopted to meet State or Federal regulations. Vehicle activity changes in response to population growth, national and regional economic conditions, changes in demand for off-road engines and the products they are used to produce, and other factors. In this section, total vehicle activity growth rates are derived for the four categories of equipment covered by the off-road rule: airport ground support equipment, construction and mining, industrial, and oil drilling.

The DOORS database and other data cited above provided a baseline 2009 population and annual equipment activity estimates for off-road rule equipment operating in California, but very little pre-2009 data is available. Therefore, historical economic indicators were used as “growth surrogates” to model how total equipment activity changes over time. The most reliable surrogates had many continuous years of historical data and correspond well with other industry trends.

a. Historical Growth Rates

Growth rates for the OFFROAD model were originally derived by Energy & Environmental Analysis (EEA) in 1995 from a study of off-road vehicle activity by CSU Fullerton (Puri and Kleinhenz, 1994). Growth rates were based on category-specific economic indicators such as employment, expenditures, and fuel use. The growth rate for the construction and mining equipment category was later derived from U.S. construction employment data from Regional Economic Models, Inc. (ARB, 2007).

For the current inventory, staff considered GDP, fuel use, employment and other economic indicators as growth surrogates for off-road equipment activity. For each data set, annual historical data were divided by 2009 values to "standardize" the data to a baseline reference point. These standardized data were termed growth factors, with a baseline value of 1.00 in 2009:

$$\text{Growth Factor} = \frac{\text{Value from Year of Interest}}{\text{Value from 2009}}$$

The slope of a linear best-fit line through the growth factor data is the historical average annual growth rate. The growth factor for a particular year is also a scaling factor by which 2009 total activity is multiplied to estimate that year's activity. Table 14 summarizes the annual growth rates used by the OFFROAD model and the values derived using the above method in this analysis. The growth rates presented here are linear and not compound. The growth rates and data sources for each equipment category are further described in the sections below.

Table D-14: Annual Historical Growth Rates

Category	OFFROAD[1]	Proposed
Airport Ground Support	2.02%	1.78%
Construction and Mining	1.96%	1.84%
Industrial	0.63%	1.84%
Oil Drilling	0.00%	0.00%

[1]Annual growth rates estimated from best-fit line fitted to growth factors from 2000 to 2030.

i. Airport Ground Support Equipment Historical Growth

For airport ground support equipment (GSE), staff considered: California air transportation GDP from the BEA (BEA, 2010); air transportation employment from the Bureau of Labor Statistics (BLS, 2010a); jet fuel use from the California Board of Equalization (BOE, 2010); airline revenue passenger miles from the Bureau of Transportation Statistics (BTS, 2010a); and U.S. airline fuel consumption from the BTS

(BTS, 2010b). GDP data is only available through 2007, which yields an overly optimistic growth rate since it does not include the recent recession. California air transportation employment data is only available since 1990 and a larger data set is necessary to derive historical average growth. Additionally, air transportation employment may not be an appropriate surrogate since contracting employment as a result of airline mergers or other factors may not directly relate to airport GSE activity. BOE California jet fuel data is only available since 2003. Annual airline passenger miles have increased steadily since the 1960s, but these data show very little variation in response to economic conditions. Therefore, airline fuel consumption, with an annual growth rate of 1.78% from 1977 to 2009, was selected as the best surrogate for airport GSE activity. Figure 15 shows the U.S. airline fuel consumption growth factors and best fit line, which is the historical growth trend.

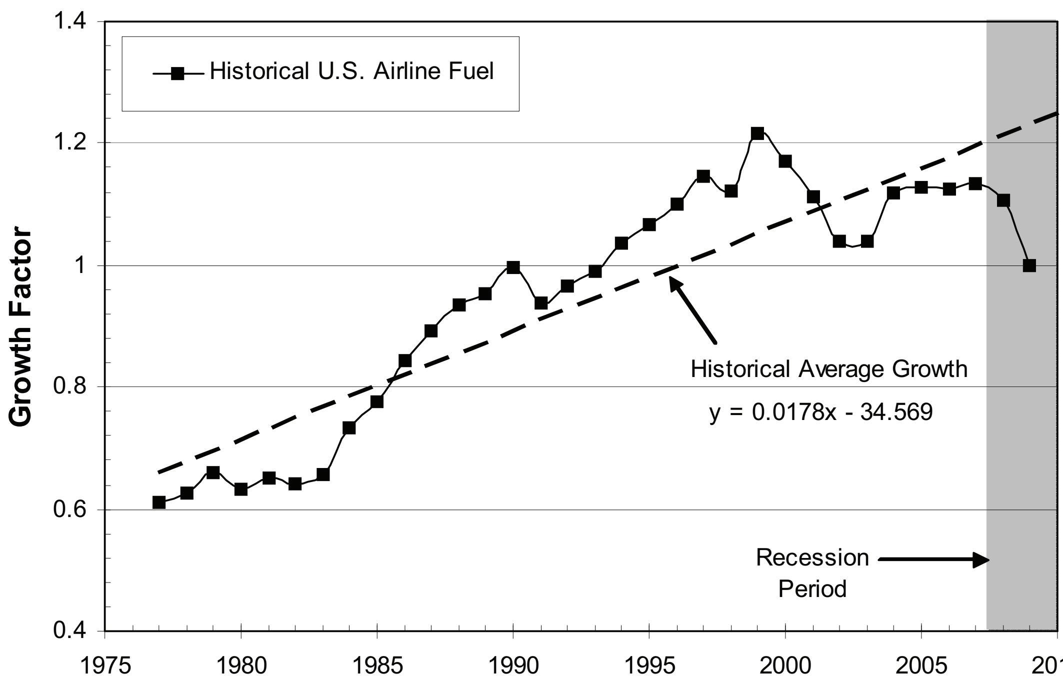


Figure D-15: Growth Factors and Best Fit Line for U.S. Airline Fuel Consumption from 1977 to 2009

ii. Construction and Mining Equipment Historical Growth

For the construction and mining sector, staff considered: U.S. Census Bureau data on the annual value of construction put in place (Census Bureau, 2010); California construction GDP from the BEA (BEA, 2010); construction sector diesel fuel sales published by the Energy Information Administration (EIA, 2010); and construction employment data from the BLS (BLS, 2010a). Data for U.S. construction value put in place were not California-specific and were only available since the early 1990's.

Historical data for California construction GDP was only available through 2008, thus its growth rate does not fully capture the effects of the recent recession. Construction sector fuel sales showed significant annual variation that is not seen in any of the other construction surrogates.

Due to the limitations described above, California construction sector employment was chosen as the best surrogate for construction and mining equipment growth. Construction employment in California had an annual growth rate of 1.84% between 1970 and 2009. California employment figures were only available since 1990, so employment from 1970 to 1989 was predicted using U.S. construction employment data and the average percentage of U.S. construction employment within California during 1990 to 2009 (10.9 ± 1.1%). Figure 16 shows the construction employment growth factors and best fit line.

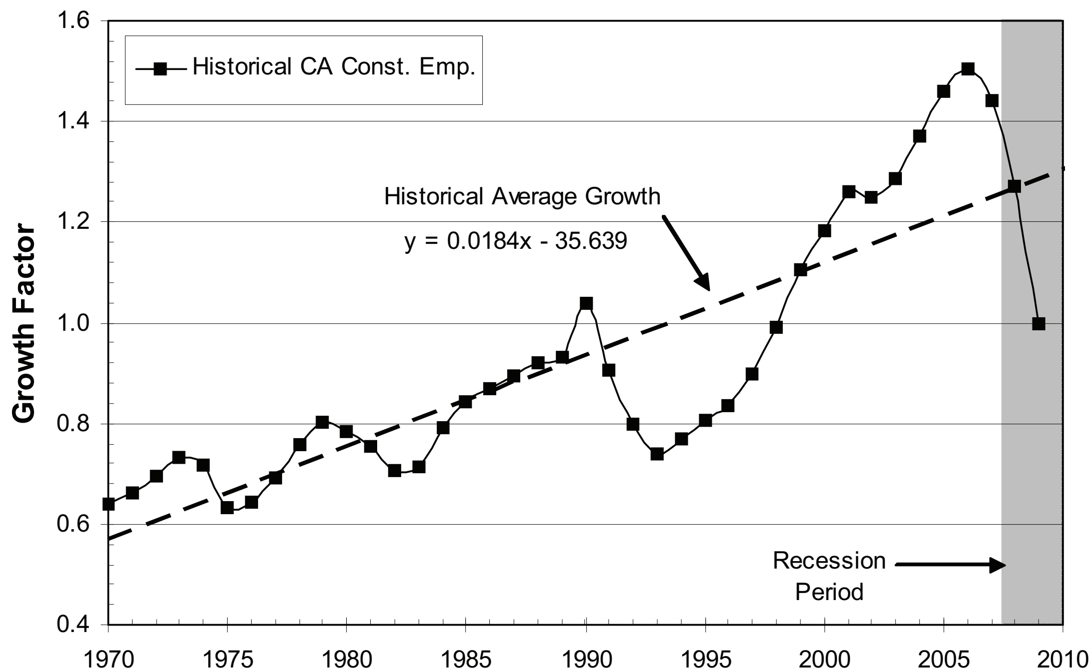


Figure D-16: Growth Factor Data and Best Fit Line for California Construction Employment from 1970 to 2009

Staff also considered California employment in the following construction occupations from the BLS: paving, surfacing, and tamping equipment operators; pile-driver operators; and operating engineers and other construction equipment operators. These occupations are more commonly associated with the use of diesel construction equipment than other construction professions (e.g. carpenters, etc.). However, unlike the other economic and employment indicators considered above, occupation-specific data were only available for one month during each year since 1999, which was insufficient to determine a historical annual growth rate. Furthermore, the month for which data was reported varied from year to year.

BLS employment projections suggested total employment in the three occupations above wiould increase 1.8% per year between 2008 and 2018 (BLS, 2010b). This value was in good agreement with the historical growth rate derived from total California construction employment.

iii. Industrial Equipment Historical Growth

Staff looked at the top industrial sector fleets reported in the DOORS database and determined that many of the fleets were closely tied to the construction industry. For this reason, and limited economic data for the industrial sector as a whole, the growth rate derived for the construction and mining category (1.84%) was also used for industrial equipment. Since industrial processes often support the construction sector, it is reasonable to assume the industrial sector mirrors construction activity.

iv. Oil Drilling Equipment Historical Growth

For the oil drilling equipment category, staff considered: California oil and gas extraction GDP from the BEA (BEA, 2010); oil company diesel fuel use published by the EIA (EIA, 2010); California rotary rig counts from Baker Hughes (Baker Hughes, 2010); and California oil and gas extraction employment from the BLS (BLS, 2010a). Oil and gas extraction GDP values were only available through 2007 and show slightly positive growth. Oil company fuel use was only available through 2008 and had a slightly negative growth rate. Oil and gas extraction employment was only available since 1990 and yielded virtually no growth over the past two decades. Rotary rig counts in California have declined since 1970, but these data did not include the horsepower or size of the equipment. All of these datasets showed major peaks and valleys over the past two decades. Considering the limitations of all the data and since no clear trend in either the positive or negative direction is exhibited, the average annual growth rate for oil drilling equipment in California was assumed to be zero.

b. Forecasting

Many of the economic indicators considered in this analysis peaked in 2006 and dropped each year through 2009 as a result of the recent recession. Preliminary data and economic forecasts suggested 2010 values will be lower than 2009. To forecast total equipment activity following the recession, staff developed three recovery scenarios to encompass the possible rate of growth (“fast”, “slow”, and “average”). In all three recovery scenarios, growth does not begin until 2011.

The fast recovery scenario assumed that total activity would return to historically average levels in 2017 and then grow at the historical average rate. A return to anticipated activity by 2017 was based on historical recovery rates and accounts for the idea that California is likely to recover more slowly from this recession than the country as a whole. Additionally, the Congressional Budget Office forecast suggested that real gross domestic product at a nationwide level will converge with potential gross domestic product trends no later than 2015. Coupling this forecast with the assumption that California's recovery will lag the nation by several years yielded the 2017 recovery date assumed for the fast recovery scenario.

For the slow recovery scenario, staff assumed that activity would be permanently depressed relative to historical levels, but continue to grow at the average historical growth rate beginning in 2011. The average scenario is the average of the fast and slow scenarios. Figure 17 illustrates the fast, slow, and average recovery scenarios through 2014 for a generic equipment type.

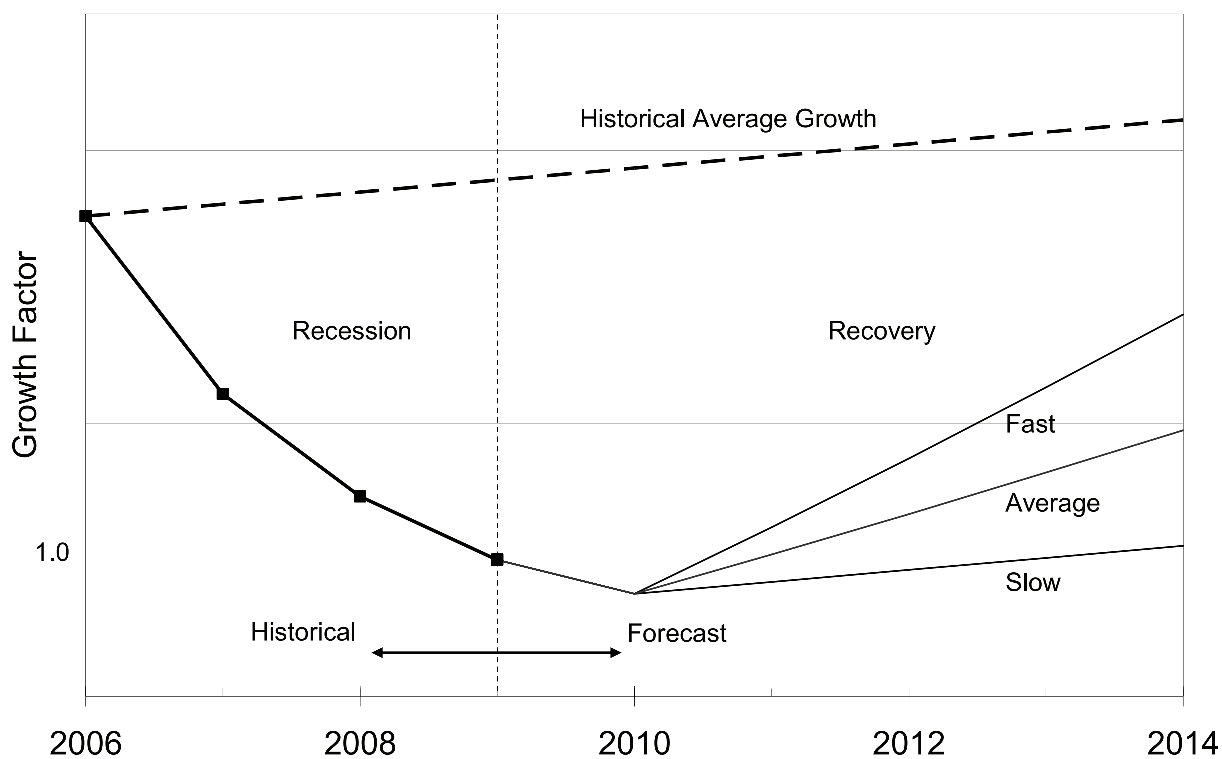


Figure D-17: Historical Average Growth (Best Fit Line to Historical Data) and ARB Fast, Slow, and Average Recovery Scenarios

Separate recovery scenarios were made for the short-term through 2014 and for the long-term from 2015 to 2030. On the short-term, staff determined that the average recovery scenario was most appropriate. Given the uncertainty with economic forecasts, the average recovery scenario represented the "middle-of-the-road" path to recovery. Staff modeled the average recovery scenario from the historical average growth rates derived for each equipment category except oil drilling equipment, which has a growth rate of zero. For the long-term growth projections, staff compared the fast, slow, and average recovery scenarios with published economic forecasts. Based on these comparisons, different long-term recovery scenarios were derived for each equipment type.

i. Airport Ground Support Equipment Forecast

To forecast short-term growth in the airport GSE sector, staff modeled the average recovery scenario from the U.S. airline fuel data used to derive the historical growth rate. The U.S. airline fuel forecast from the Federal Aviation Administration (FAA) predicted fuel consumption in 2010 will be 3% lower than 2009 (FAA, 2010). Therefore, the 2010 growth factor is assigned a value of 0.97. Figure 18 shows historical U.S. airline fuel growth factor data from 2006 to 2009, the historical average growth rate, and the short-term recovery scenario through 2014.

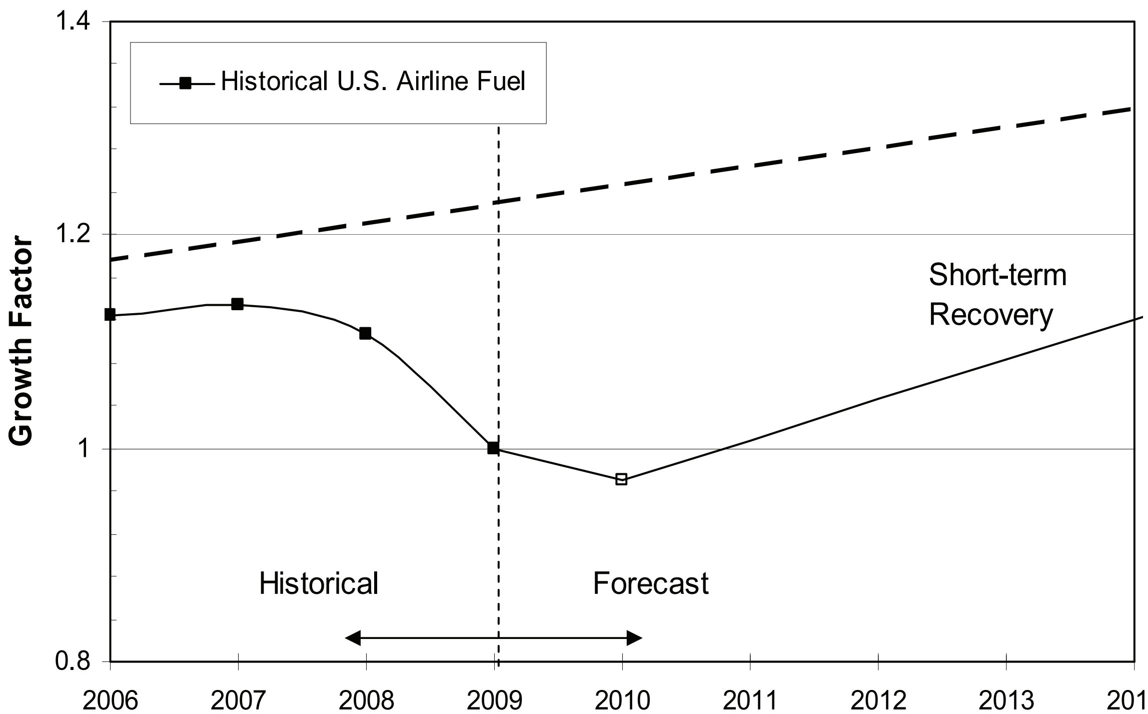


Figure D-18: Historical U.S. Airline Fuel Consumption Growth Factors, Historical Average Growth Rate, and ARB Short-term Recovery Scenario

Staff then compared the fast, slow, and average recovery scenarios to the FAA U.S. airline fuel forecast to generate a long-term recovery scenario. Based on this comparison, the average recovery scenario was selected through the year 2030. Figure 19 shows historical U.S. airline fuel consumption growth factor data from the years 1977 through 2009 and the ARB long-term recovery scenario. As shown in Fig. 19, the long-term scenario recovers more rapidly through 2017 and then grows at the historical average growth rate through 2030.

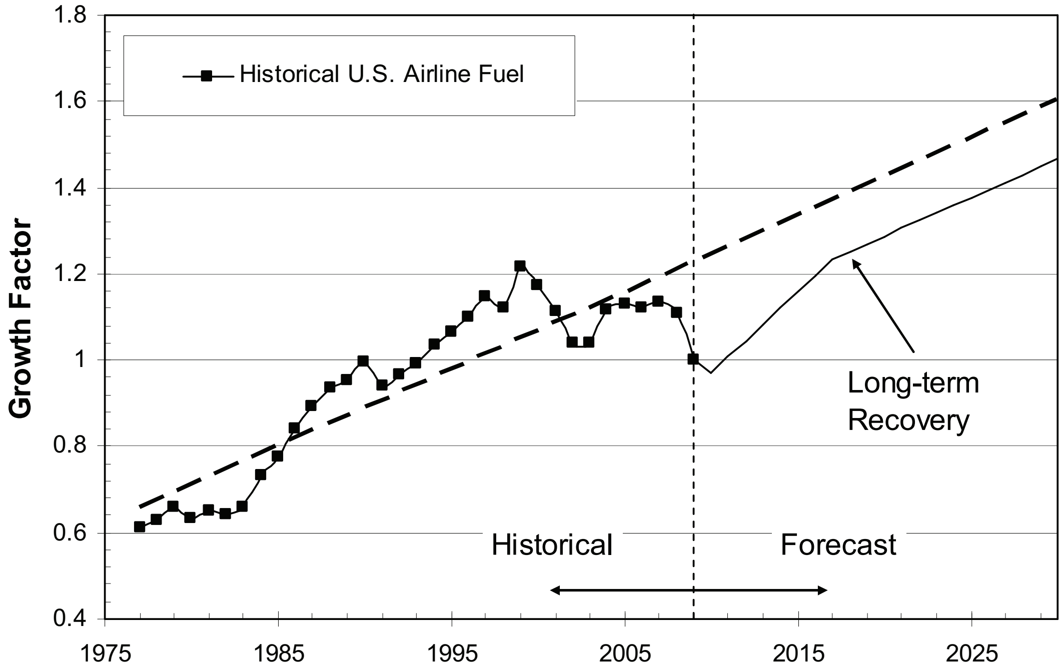


Figure D-19: Historical U.S. Airline Fuel Consumption, Historical Average Growth Rate, and ARB Long-term Recovery Scenario

ii. Construction and Mining Equipment Forecast

To forecast growth in the construction and mining sector, staff modeled the average recovery scenario from the California construction employment data used to develop the historical growth rate. Forecasts of California construction employment from the UCLA Anderson School of Management (UCLA, 2010), Beacon Economics (Beacon, 2010), and University of the Pacific Eberhardt School of Business (UOP, 2010) predicted construction employment in 2010 is about 10% lower than 2009. For this reason, the 2010 growth factor was assigned a value of 0.90. Figure 20 shows historical CA construction employment from 2006 to 2009, the historical average growth rate, and the short-term recovery scenario through 2014.

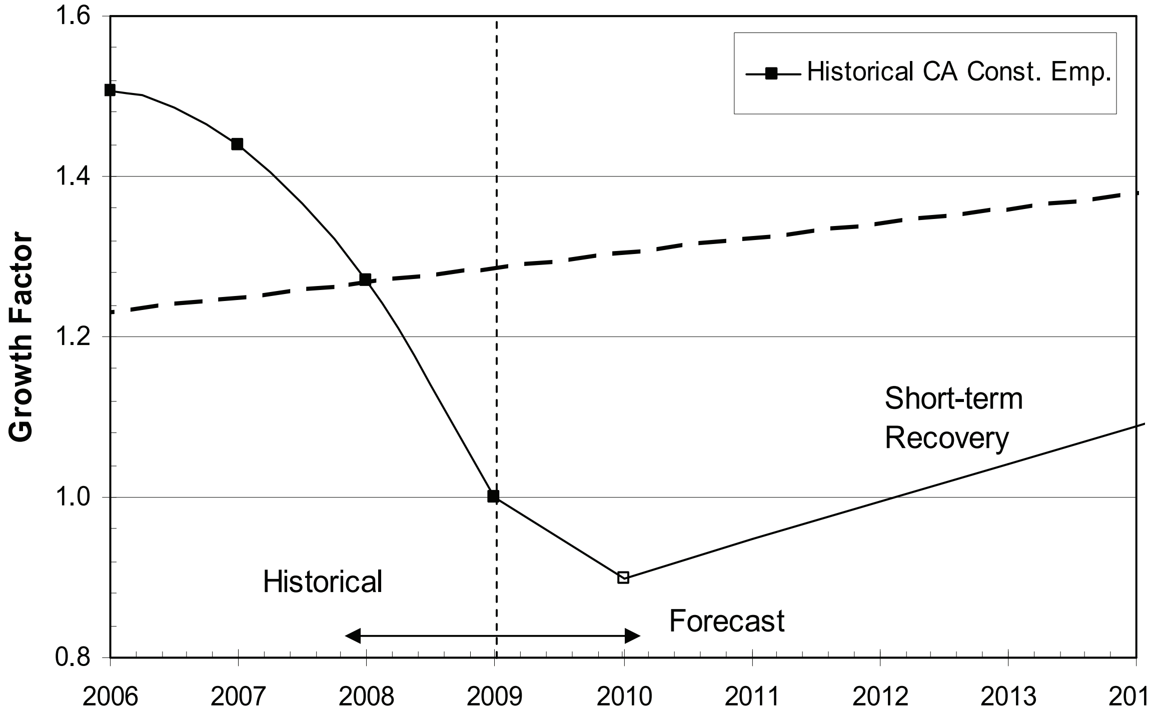


Figure D-20: Historical CA Construction Employment Growth Factors, Historical Average Growth Rate, and ARB Short-term Recovery Scenario

Staff then compared the fast, slow, and average recovery scenarios to the published construction employment forecasts referenced above. Based on this comparison, the long-term recovery scenario was modeled by extending the short-term recovery scenario trend line past 2014 until it reached the historical average growth trend. The long-term recovery scenario intersects the historical average growth trend in 2024 and then follows the same growth rate as the historical average. Figure 21 shows the historical construction employment data from 1970 to 2009, construction employment forecast data, and the long-term recovery scenario through 2030.

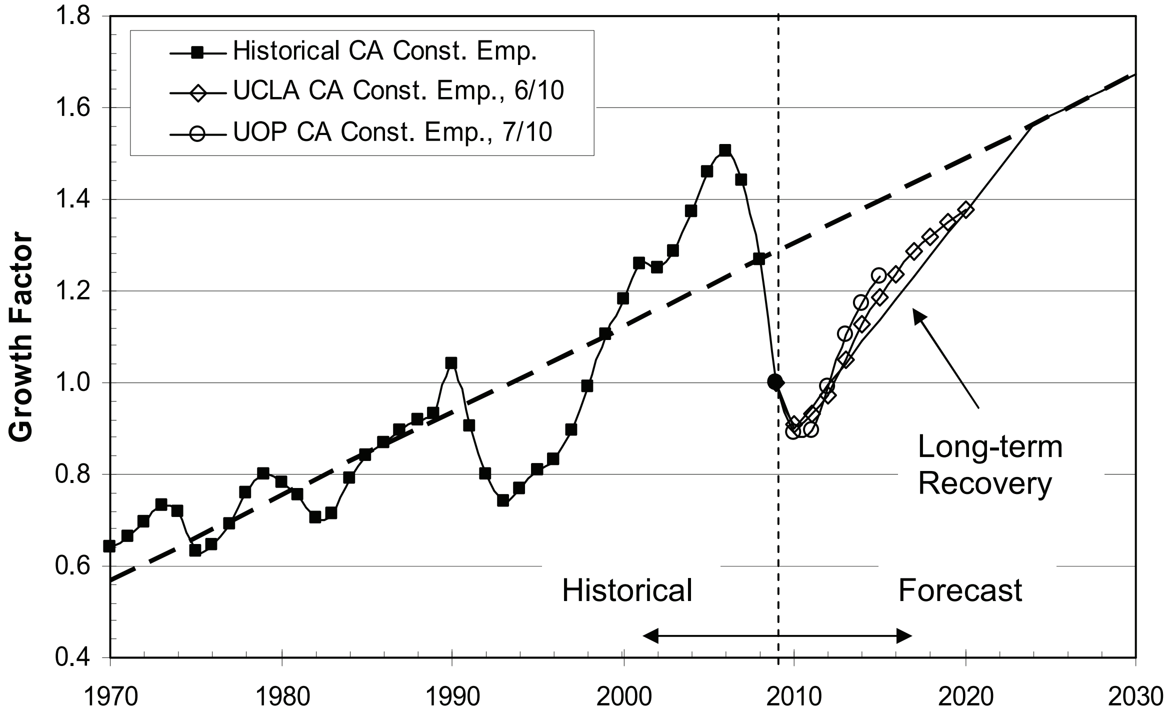


Figure D-21: Historical CA Construction Employment, Historical Average Growth Rate, Forecasts, and ARB Long-term Recovery Scenario

iii. Industrial Equipment Forecast

As described above, industrial equipment was assigned the same annual growth rate as construction and mining equipment. Forecasted growth in industrial equipment was also assumed to be the same as the construction and mining category.

iv. Oil Drilling Equipment Forecast

With an annual growth rate of 0.00%, the oil drilling category was assigned a growth factor of 1.00 for all future years.

c. Growth Factors

For each equipment category, staff derived growth factors to represent future activity. A growth factor is a scaling factor by which the baseline total activity is multiplied to arrive at a future equipment activity. The year 2009 was assigned a baseline growth factor of 1.00 for all equipment types and 2010 values were derived from economic forecasts. As described above, airport ground support equipment had a growth factor of 0.97 in 2010 and the construction and mining and industrial categories had growth factors of

0.90 in 2010. Table 15 lists the growth factors by equipment category for future years 2010 through 2030.

Table D-15: Future Growth Factors by Equipment Category for 2010 to 2030

Year	Airport Ground Support	Construction and Mining	Industrial	Oil Drilling
2009	1.00	1.00	1.00	1.00
2010	0.97	0.90	0.90	1.00
2011	1.01	0.95	0.95	1.00
2012	1.05	0.99	0.99	1.00
2013	1.08	1.04	1.04	1.00
2014	1.12	1.09	1.09	1.00
2015	1.16	1.14	1.14	1.00
2016	1.20	1.18	1.18	1.00
2017	1.23	1.23	1.23	1.00
2018	1.25	1.28	1.28	1.00
2019	1.27	1.33	1.33	1.00
2020	1.29	1.37	1.37	1.00
2021	1.31	1.42	1.42	1.00
2022	1.32	1.47	1.47	1.00
2023	1.34	1.51	1.51	1.00
2024	1.36	1.56	1.56	1.00
2025	1.38	1.58	1.58	1.00
2026	1.39	1.60	1.60	1.00
2027	1.41	1.62	1.62	1.00
2028	1.43	1.64	1.64	1.00
2029	1.45	1.65	1.65	1.00
2030	1.47	1.67	1.67	1.00

II. EMISSIONS INVENTORY RESULTS

A. Statewide

Overall, the updated NOx emissions, PM emissions and fuel consumption estimates are lower statewide and for the South Coast and San Joaquin Valley Air Basins. Most of the reductions can be attributed to changes in the data sources used as inputs for the model as described below:

- **Population**: Statewide population estimates in the updated inventory were derived from the DOORS database (DOORS, 2010). The DOORS population estimates were lower than the nationwide survey data used to develop population estimates for the original OFFROAD inventory.
- **Activity**: Annual activity data in the updated inventory were derived from Assembly Bill 8 2x reporting data (AB 8 2x, 2009), which were specific to California fleet activity. These activity values were lower than the activity values used for the original model.
- **Load Factor**: Updated load factor data for the new inventory were derived from ARB and manufacturer provided data. These load factor data were lower than the values derived from Power Systems Research data (Power Systems Research, 1996) used in the original inventory.

Table 13 shows statewide NOx emissions estimates for the unregulated 'baseline' inventory and remaining emissions estimates for the proposed regulation. The estimated benefits of the regulation project a decrease in emissions by 2% in 2014, 13% in 2020, and 17% in 2023. Figure 22 shows a year-by-year comparison of NOx emissions for baseline and proposed-regulation scenarios.

Table 1: Statewide NOx Emissions – Baseline and With-Regulation (tons per day)

	2009	2014	2020	2023
NOx Baseline	79	76	65	52
NOx With Proposed Regulation	79	75	57	44
Benefits of Regulation	0	1	8	9
Percent Reduction	0%	2%	13%	17%

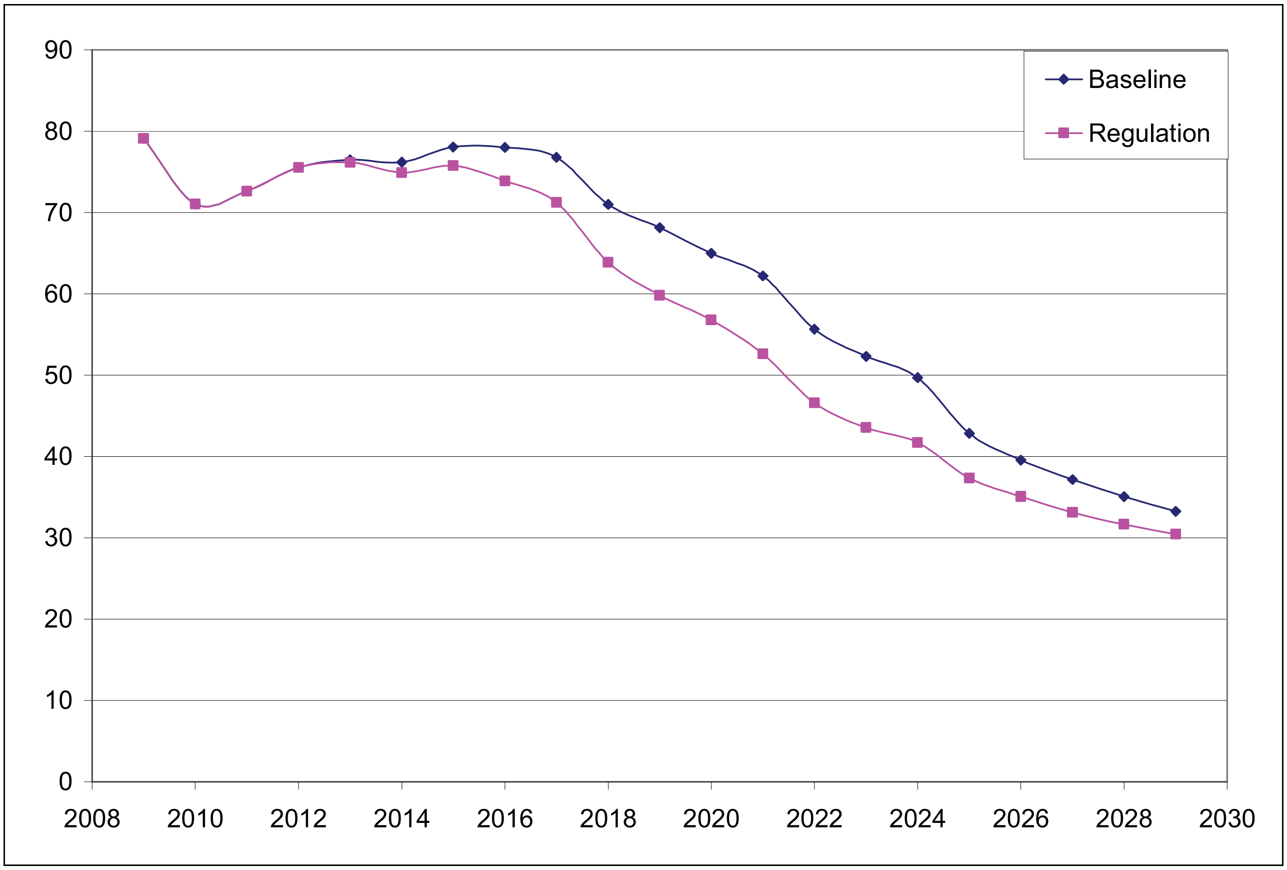


Figure 22: Statewide NOx Emissions – Baseline and With-Regulation (tons per day)

Table 15 shows statewide PM 2.5 emissions estimates for the unregulated 'baseline' inventory and remaining emissions estimates for the proposed regulation. The estimated benefits of the regulation project a decrease in emissions by 2% in 2014, 16% in 2020, and 21% in 2023. Figure 23 shows a year-by-year comparison of PM 2.5 emissions for baseline and proposed-regulation scenarios.

Table 2: Statewide PM 2.5 Emissions – Baseline and With-Regulation (tons per day)

	2009	2014	2020	2023
PM 2.5 Baseline	3.7	3.6	2.9	2.2
PM 2.5 With Proposed Regulation	3.7	3.5	2.4	1.7
Benefits of Regulation	0.0	0.1	0.5	0.5
Percent Reduction	0%	2%	16%	21%

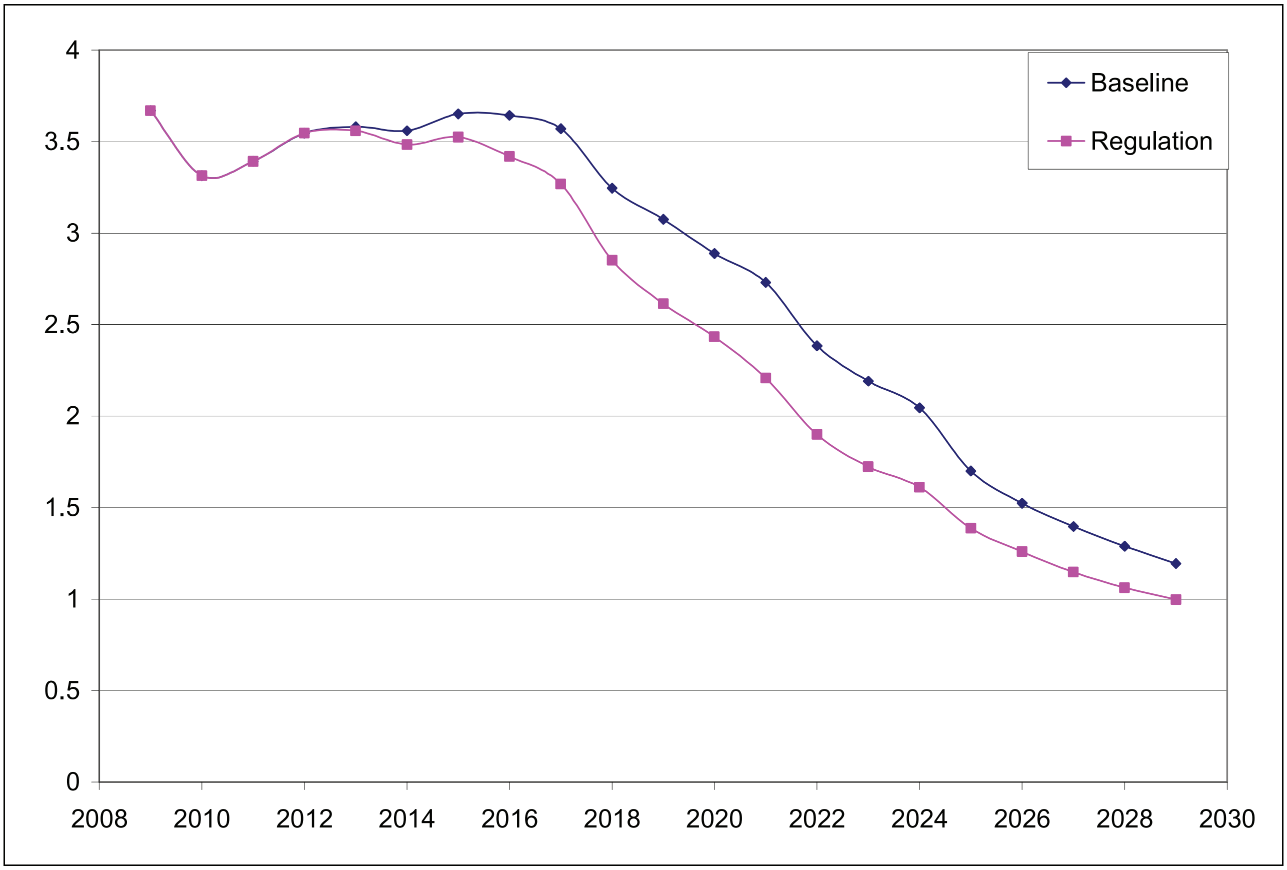


Figure 23: Statewide PM 2.5 Emissions – Baseline and With-Regulation (tons per day)

Table 16 shows original fuel consumption estimates from the OFFROAD model compared to new estimates. OFFROAD estimates 990 million gallons per year in 2009 and increases to 1,170 million gallons per year by 2023. Updated estimates are 220 million gallons per year in 2009 and increases to 330 million gallons per year by 2023. Overall updated fuel estimates are about one fourth original values.

Table 3: Statewide Fuel Consumption - OFFROAD and Update (million gallons per year)

OFFROAD	**2009**	**2014**	**2020**	**2023**
Fuel Consumption	990	1,020	1,130	1,170
UPDATE	**2009**	**2014**	**2020**	**2023**
Fuel Consumption	220	240	300	330

While there is some information available for off-road fuel sales there is no single source that breaks out sales for In-Use Off-Road Rule categories. Sources that do report fuel sales for subsets of off-road equipment have uncertainty as trends in the data do not match industry activity. The updated emissions inventory presented here reflects the best available data provided directly by California fleets that wasn't available at the time of the original rulemaking. Fuel consumption now agrees more closely to other fuel based estimates.

Figure 23 below shows the 2007 fuel consumption estimates from the OFFROAD model and the updated estimate for calendar year 2009. Also presented are the range of estimates provided by the Associated General Contractors (AGC) and the American Rental Association (ARA).

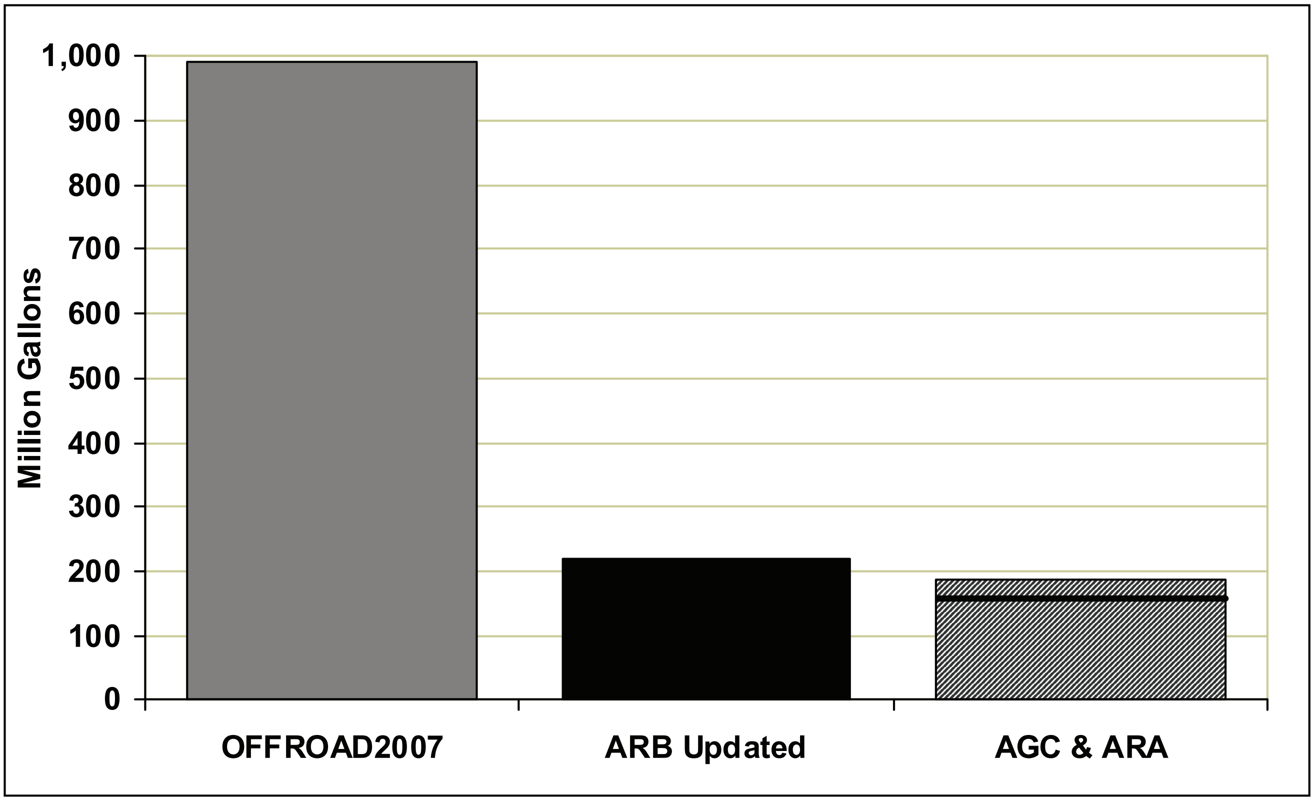


Figure 23: 2009 Statewide Fuel Consumption Comparison – OFFROAD, Update and Industry

B. Air Basins – Updated South Coast and San Joaquin Valley

In order to estimate regional emissions, staff allocated statewide estimates to the various air basins in the state. Allocation factors in the updated inventory have been changed since the original rulemaking inventory (see I-D-1). Construction and mining equipment allocations were updated with county-specific population growth projections; industrial equipment allocations were updated with employment data from the 'Goods-Producing' and 'Manufacturing' divisions; airport ground support equipment and oil drilling remained the same as they were in OFFROAD. These changes resulted in different air basin emissions relative to statewide estimates (see Tables 10 and 12).

Tables 17 and 18 show updated emissions estimates for the South Coast and San Joaquin Valley Air Basins, respectively. A comparison between baseline and with-regulation scenarios is shown for PM 2.5, NOx, and ROG.

Table 4: Emissions Inventory for South Coast (tons per day)

South Coast						
Calendar Year	PM2.5		NOx		ROG	
	Baseline	Regulation	Baseline	Regulation	Baseline	Regulation
2009	1.3	1.3	28	28	2.5	2.5
2014	1.3	1.3	28	28	2.6	2.5
2020	1.1	0.9	25	22	2.4	2.1
2023	0.8	0.6	19	16	2.0	1.7

Table 5: Emissions Inventory for San Joaquin Valley (tons per day)

San Joaquin Valley						
Calendar Year	PM2.5		NOx		ROG	
	Baseline	Regulation	Baseline	Regulation	Baseline	Regulation
2009	0.5	0.5	12	12	1.0	1.0
2014	0.6	0.6	14	13	1.2	1.2
2020	0.6	0.5	14	12	1.3	1.2
2023	0.5	0.4	11	9	1.2	1.0

C. State Implementation Plan (SIP) and Preserving Benefits

As discussed in the staff report the combined impact of the recession with amended regulations on both on-road and off-road inventories will provide essentially the same cumulative remaining emissions levels between 2011 and 2023 to what the adopted Rules would have achieved without the recession. In addition, the combined margin for trucks/buses and off-road equipment is minimized, providing maximum relief while still meeting SIP legal obligations. For a more detailed discussion of these issues please refer to Section H of Appendix D in the “Proposed Amendments to the Truck and Bus Regulation, the Drayage Truck Regulation and the Tractor-Trailer Greenhouse Gas Regulation” staff report.

Attachment A

III. Off-Road Simulation Model (OSM)

A. Introduction and Overview

ARB staff have developed the Off-road Simulation Model (OSM) to provide future-year statewide vehicle populations for use in the off-road emissions inventory and to evaluate various alternative amendments to the current off-road regulation. OSM utilizes the California off-road fleet data submitted to DOORS; from these fleet data, future year fleet composition is projected by OSM on a fleet-by-fleet basis. OSM consists of PHP code (pages) that manipulate DOORS fleet data in a separate OSM database (both DOORS and OSM database are in MySQL). The OSM vehicle population projections used in the Offroad Inventory and the analyses within this document are based upon vehicles registered in DOORS as of September 23, 2010.

This section provides a brief description of the user interface and operation, OSM logic, followed by a more detailed description of OSM and how the vehicle populations were derived.

[Note: Both the Offroad Inventory and OSM use 2009 as a beginning year, however OSM designates actions taken by fleets from April 1, 2009 through March 31, 2010 as “2010”. Similarly, for succeeding years, the designation of a “regulatory year” represents the actions taken in the twelve months leading up to March 1st of the year.]

i. OSM User Interface and Operation

A user can run OSM from the user interface shown in Figure 1. As can be seen, the user has only a single button to click to run either the proposed regulation amendments or any of three alternatives. Clicking a button will cause OSM to populate tables within the OSM database with projected future year vehicle populations. When this operation is complete a cost table will be displayed as shown in Figure 2. The raw vehicle and fleet data that are used as inputs to the inventory are contained within several tables within the OSM database as shown in Figure 3.

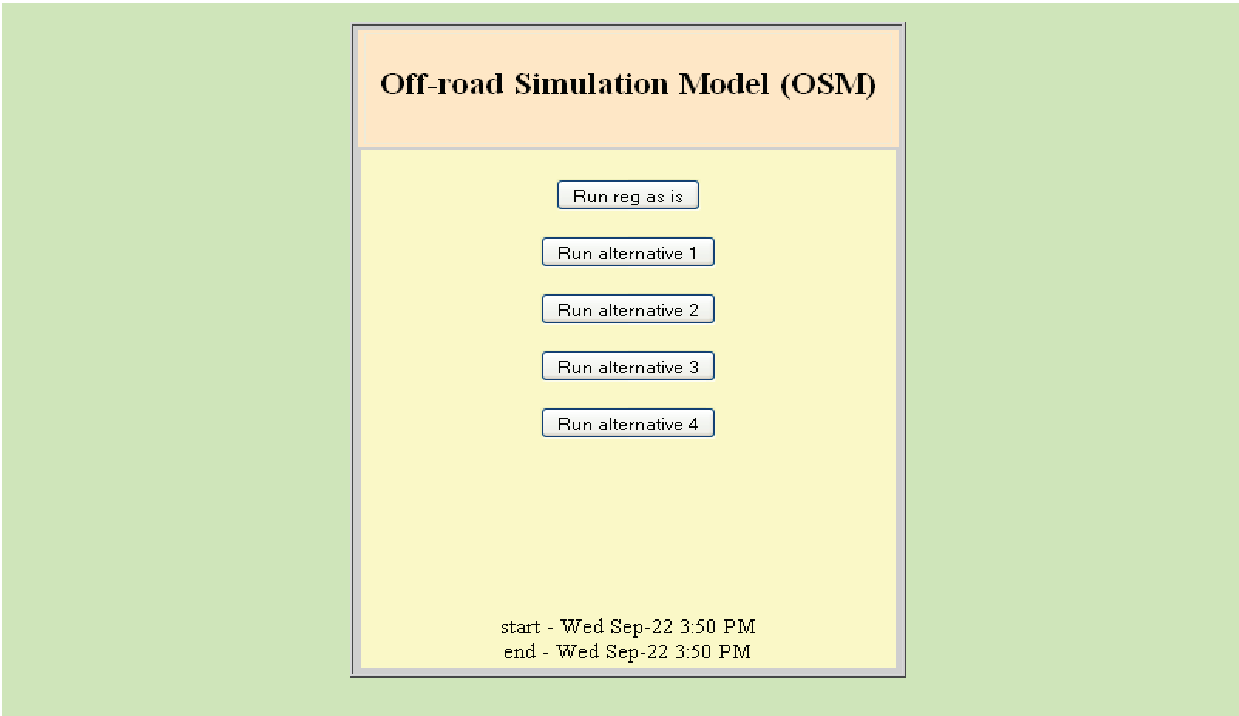
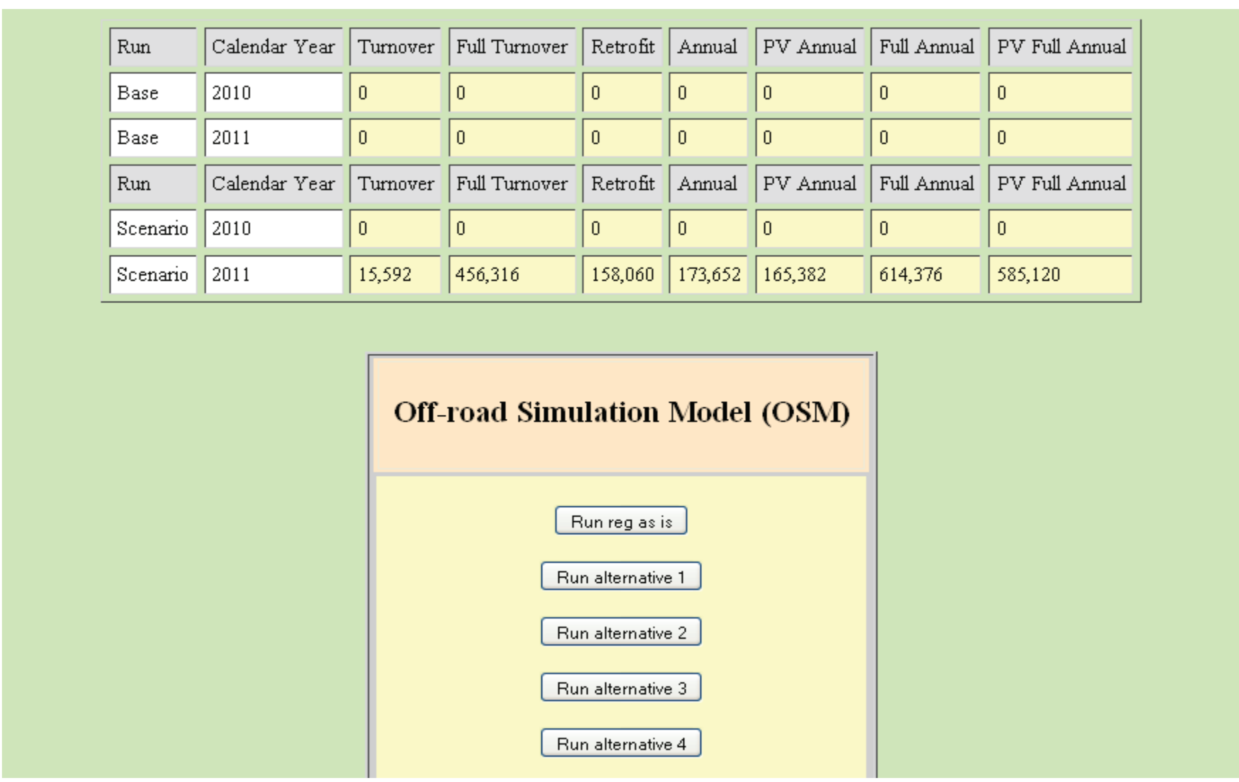


Figure 1: OSM User Interface



Run	Calendar Year	Turnover	Full Turnover	Retrofit	Annual	PV Annual	Full Annual	PV Full Annual
Base	2010	0	0	0	0	0	0	0
Base	2011	0	0	0	0	0	0	0
Run	Calendar Year	Turnover	Full Turnover	Retrofit	Annual	PV Annual	Full Annual	PV Full Annual
Scenario	2010	0	0	0	0	0	0	0
Scenario	2011	15,592	456,316	158,060	173,652	165,382	614,376	585,120

Figure 2: OSM Cost Table

```
mysql> use osm1085
Database changed
mysql> show tables;
+--------------------------------------+
| Tables_in_osm1085                    |
+--------------------------------------+
| osm_fleet_base_costs                 |
| osm_fleet_base_costs_cumulative      |
| osm_fleet_scenario_costs             |
| osm_fleet_scenario_costs_cumulative  |
| osm_vehicle                          |
| osm_vehicle_base                     |
| osm_vehicle_base_costs               |
| osm_vehicle_scenario                 |
| osm_vehicle_scenario_costs           |
| zeconomic_recovery_factor            |
| zengine_index                        |
| zengine_target                       |
| zequipment_class                     |
| zretrofit_cost                       |
| ztarget_1delay                       |
| ztarget_2delay                       |
| ztarget_2delay_noshift               |
| ztarget_3delay                       |
| ztarget_4delay                       |
| ztarget_5delay                       |
| ztier4_premium_cost                  |
| zvehicle_average_age                 |
| zvehicle_cost                        |
+--------------------------------------+
23 rows in set (0.01 sec)

mysql>
```

Figure 3: OSM Database Tables

ii. OSM Logic

As mentioned the user may run an OSM simulation of the existing off-road regulation or an amended off-road regulation by clicking a button in the user interface. Each button will load the specific parameters for that simulation.

Each simulation will consist of a “base” run and a “scenario” run. The base run models fleet behavior without any regulatory constraints, but only with actions that are consistent with usual business practices. On the other hand, the scenario run includes both actions taken to meet usual business practices but also additional actions required by the regulation or amended regulation. Both the scenario and base runs are needed for each fleet to see the change in fleet composition and calculate the associated costs of compliance.

Under either the scenario or base run, OSM starts with the fleet information from fleets reported in DOORS as of September 23, 2010 and then copies the fleet information for an individual fleet into the next regulatory year. Then, OSM starts modifying this fleet with turnover and retrofits. As mentioned, depending on an input parameter OSM may follow the constraints given in the regulation while turning over and retrofitting vehicles or not, but in either case, OSM will modify the fleet over time. OSM then projects the fleet into the next regulatory year and repeats the process for the fleet through regulatory year 2030. That process is then repeated for each fleet until all fleets have been modified. At the conclusion of an OSM run, database tables will contain data that reflects the changes to the fleet composition.

To summarize the programming logic of the preceding paragraph, in the most general terms, OSM consists of processing each fleet in DOORS in turn containing a nested loop processing each year in turn which contains another nested loop processing turnover and retrofit actions taken on each individual vehicle in turn. The process is illustrated in Figure 4.

Figure 5 is a flowchart showing OSM logic in more detail. As shown, OSM includes pages that initialize variables, gets fleet information, creates new regulatory years, gets the fleet targets, gets the fleet averages, sets compliance flags, sets the pool of eligible vehicles, modifies the fleet table with turnover and retrofits, writes the results to the database, and performs other actions as needed. More details are given in the following section, OSM Formulas.

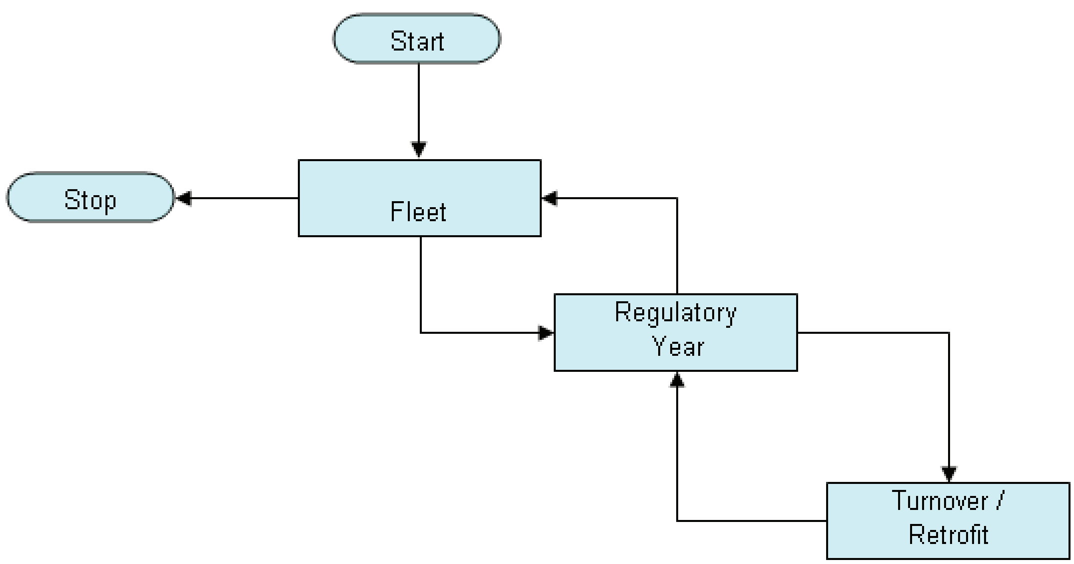


Figure 4: Summary OSM Flowchart

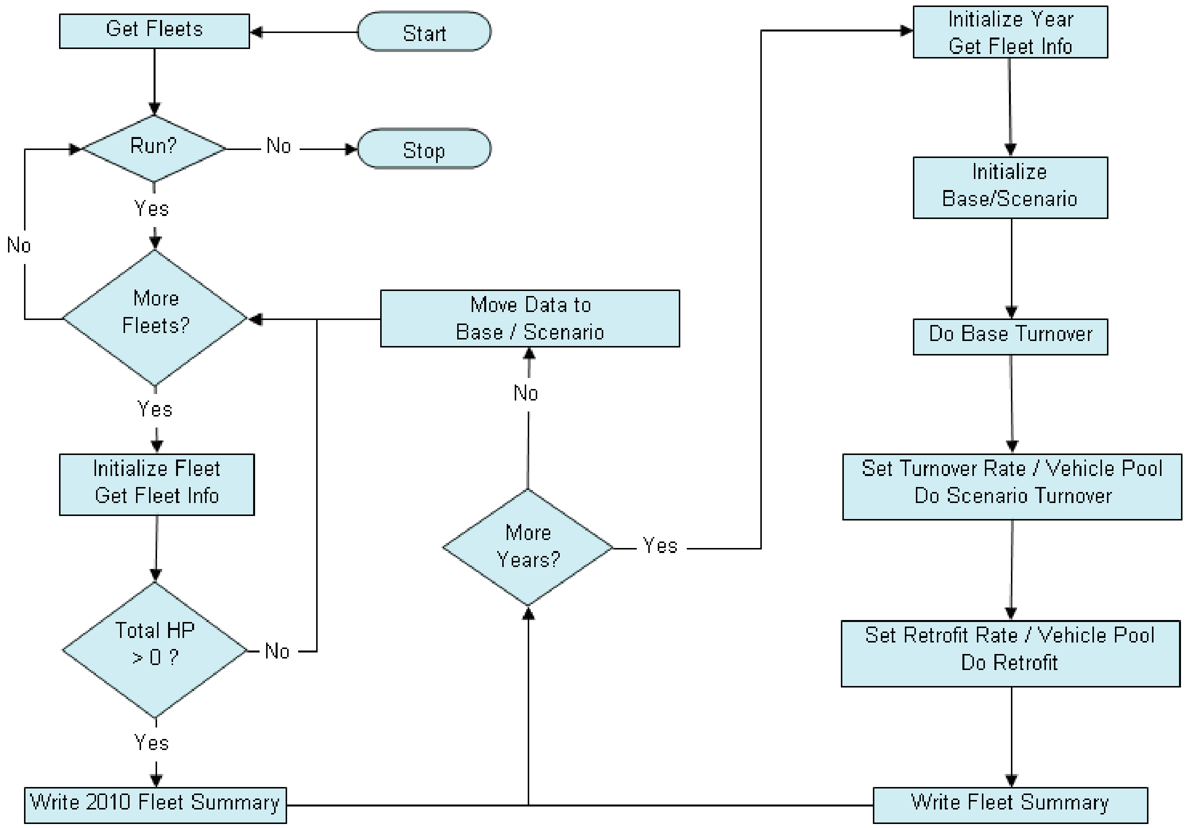


Figure 5: OSM Flowchart

iii. OSM Formulas

OSM first initializes the fleet variables (those variables that must be reset for each fleet but shouldn't be initialized each year, such as beginning turnover credit, retrofit credit, etc.), then initializes the variables that will change from year to year, such as the turnover needed and the retrofit needed for a given year. After all the variables are initialized, the fleet information is retrieved from a September 23, 2010 copy of DOORS as well as any early credit and then the fleet is stored in the osm_vehicle table for the regulatory year 2010. Then the osm_vehicle table is supplemented with additional information from lookup tables such as vehicle average age.

Low-use vehicles are included in the information retrieved from DOORS but are excluded from turnover and retrofit. Low-use vehicles are included in the emissions estimates and should reasonably model actual emissions since low-use vehicles that remain unchanged in the fleet really represent replacement with newer, but still relatively old, vehicles that become low-use. This would be the case for vehicles that have more than 12,000 hours since deterioration in the emissions rate is truncated beyond that point and hence emissions remain the same. [However, the age distribution will be distorted by low-use vehicles in later years since they will not reflect replacement with newer model vehicles.]

a. get_fleet_info.php

The get_fleet_info.php page calculates total horsepower and fleet age for the current fleet excluding low-use vehicles. The fleet total horsepower is simply the sum of the horsepower of all vehicles in the fleet excluding low use vehicles.

The calculation of the total horsepower is shown in Equation 1.

Equation 1: $$TotalHorsepower = \sum_{i=1}^{n} Horsepower_i$$

Where n = Number of vehicles in the fleet excluding low use vehicles.

Fleets with missing engine information will have an indeterminate total horsepower and therefore total horsepower is set equal to zero and the fleet will not be included in the OSM calculations.

The fleet age is a measure of the horsepower weighted average age of the fleet as a whole. The fleet age is calculated in Equation 2.

Equation 2: $$FleetAge = \frac{\sum_{i=1}^{n} EngineHorsepower * (CalendarYear - EngineModelYear_i)}{TotalHorsepower}$$

The fleet age in regulatory year 2010 is fixed as a measure against which turnover is controlled as explained below.

Equation 3: $$FleetAverageAge2010 = FleetAge(2010)$$

b. do_turnover_base.php

OSM will do a "base turnover" in both the base and scenario runs in the do_turnover_base.php page (additional turnover will occur under the scenario run in response to the regulatory requirements in the do_turnover.php page as discussed below). It is assumed that under normal conditions without a regulation forcing accelerated turnover, a fleet will turnover at a rate such that the fleet average age is maintained, on average, constant over time.

An adjusted 2010 fleet age incorporates the impact of an assumed economic recovery on the 2010 fleet age measure by utilizing an economic recovery factor as shown in Equation 4. In the economic recovery it is assumed that vehicles are held longer by the fleet owner than in normal economic times and therefore the fleet age would naturally become older.

Equation 4:

$$FleetAgeAdjusted2010 = \frac{\sum_{i=1}^{n} EngineHorsepower * (CalendarYear - EngineModelYear_i) * EconomicRecoveryFactor}{TotalHorsepower}$$

An excerpt from the Economic Recovery by Year is shown in Table 1.

Table 1: Excerpt from Economic Recovery Factor Table

Equipment Class	2010	2011	2012	2013	2014	2015
Airport Ground Support	1.00	1.01	1.02	1.03	1.03	1.03
Construction and Mining	1.00	1.08	1.13	1.18	1.22	1.24
Industrial	1.00	1.12	1.20	1.26	1.28	1.29
Oil Drilling	1.00	1.00	1.00	1.00	1.00	1.00

As discussed, the fleet age is maintained at the original 2010 fleet age with adjustments to account for economic recovery. Each year, if no vehicles were turned over, the fleet would age by one year in each succeeding year; thus, to maintain a constant fleet age in each successive year, vehicles are turned over one at a time and after each vehicle is turned over the fleet age is recalculated. Once the fleet age is no longer greater than the original adjusted 2010 fleet age (plus a delta discussed below), turnover stops. This is illustrated in the Base Turnover Flowchart shown in Figure 6.

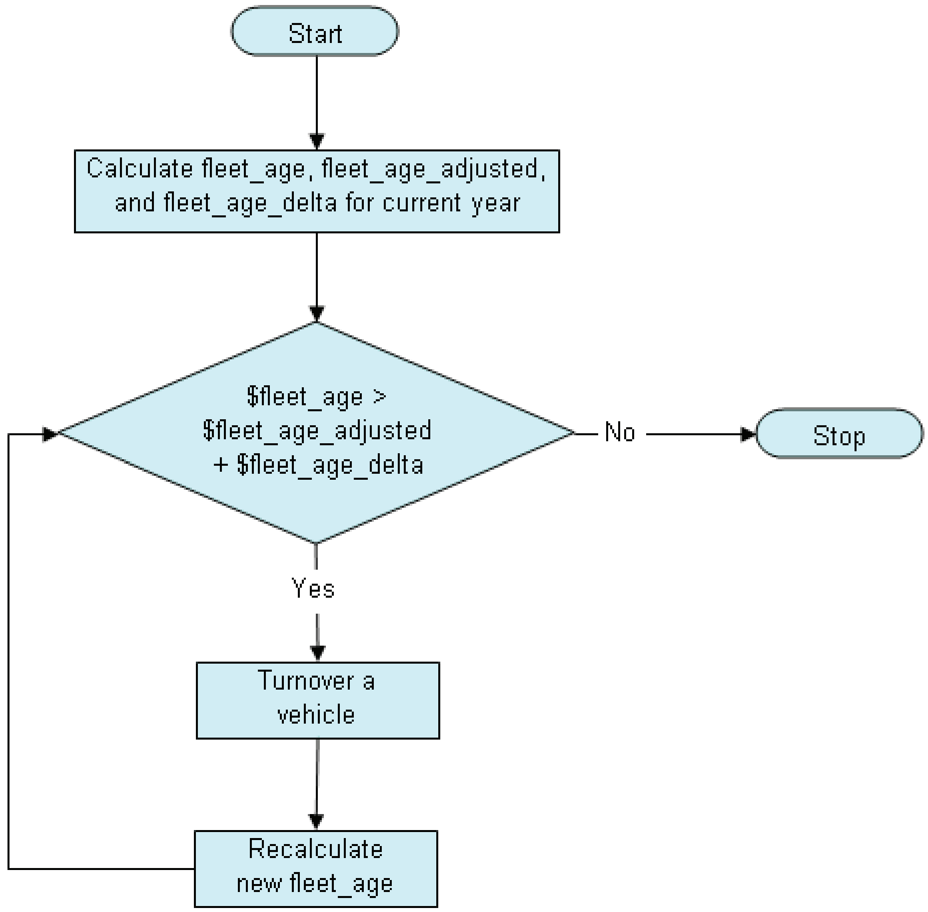


Figure 6: Base Turnover Flowchart

Since discrete vehicles will be turned over, the fleet age will virtually always overshoot the targeted fleet adjusted age resulting in a slightly younger fleet than required. Therefore a fleet age delta is calculated according to Equation 5 which is then added to the target fleet age when determining when to stop turnover as shown in equation 6. Over a period of time, the average fleet age over all years will asymptotically approach the original 2010 fleet age.

Equation 5:

$$FleetAgeDelta_{CurrentYear} = FleetAgeDelta_{PriorYear} + FleetAge2010Adjusted - FleetAge_{CurrentYear}$$

Equation 6: $FleetAge_{CurrentYear} > FleetAgeAdjusted_{2010} + FleetAgeDelta_{PriorYear}$

For example, a fleet with a horsepower weighted, average age of 13.2 years will continue as a 13.2 year old fleet over time, with some variation due to the discrete nature of individual vehicle replacements. The turnover which keeps the fleet less than or equal to the original fleet age is called the “base turnover”.

Base turnover is illustrated in Figure 7; in this example in 2016, the fleet happens to have an average age equal to its 2010 fleet average age of 13.2 years. In 2017 if no vehicles were turned over the fleet average age would be 14.2 years; however a vehicle is turned over which reduces the average fleet age to 13.9, then another vehicle is turnover to reduce the age to 13.5, then another to reduce the age to 13.0. At this point, the average age is less than the 2010 average fleet age and therefore base turnover had been achieved.

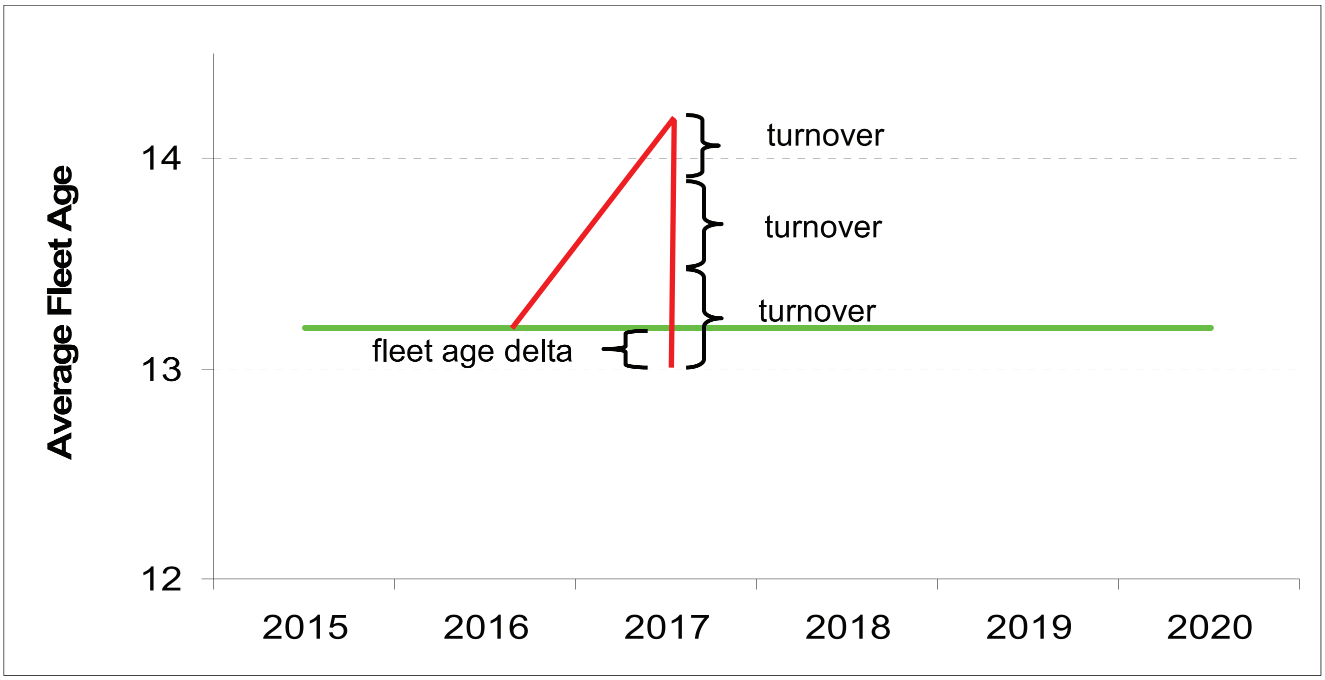


Figure 7: Base Turnover

Due to the discrete nature of vehicle turnover, the age after turnover in 2016 is less than the 2010 fleet age, the running total of that amount is call the fleet age delta. As shown in Equation 5, the fleet age delta in a given year would be the fleet age delta from the prior year plus the difference between the 2010 fleet age and the current year fleet age. So in the prior example the fleet age delta would be 13.2 – 13.0 = 0.2 years.

Since in 2017 the fleet average age was less than the 2010 fleet average age by the amount of the fleet age delta, in the next year the fleet average age is allowed to exceed the 2010 fleet average age by that amount. As shown in Figure 8, if there were no turnover in 2018 the fleet age would increase from 13.0 to 14.0; however, a vehicle is turned over which reduces the fleet average age to 13.7, and then another vehicle is turned over reducing the average age to 13.3 years old. Since the 2010 fleet average age of 13.2 plus the fleet age delta of the prior year of 0.2 years is 13.4 years, base turnover has been achieved. Note that the fleet age delta going forward is now 13.4-13.3 = 0.1 years.

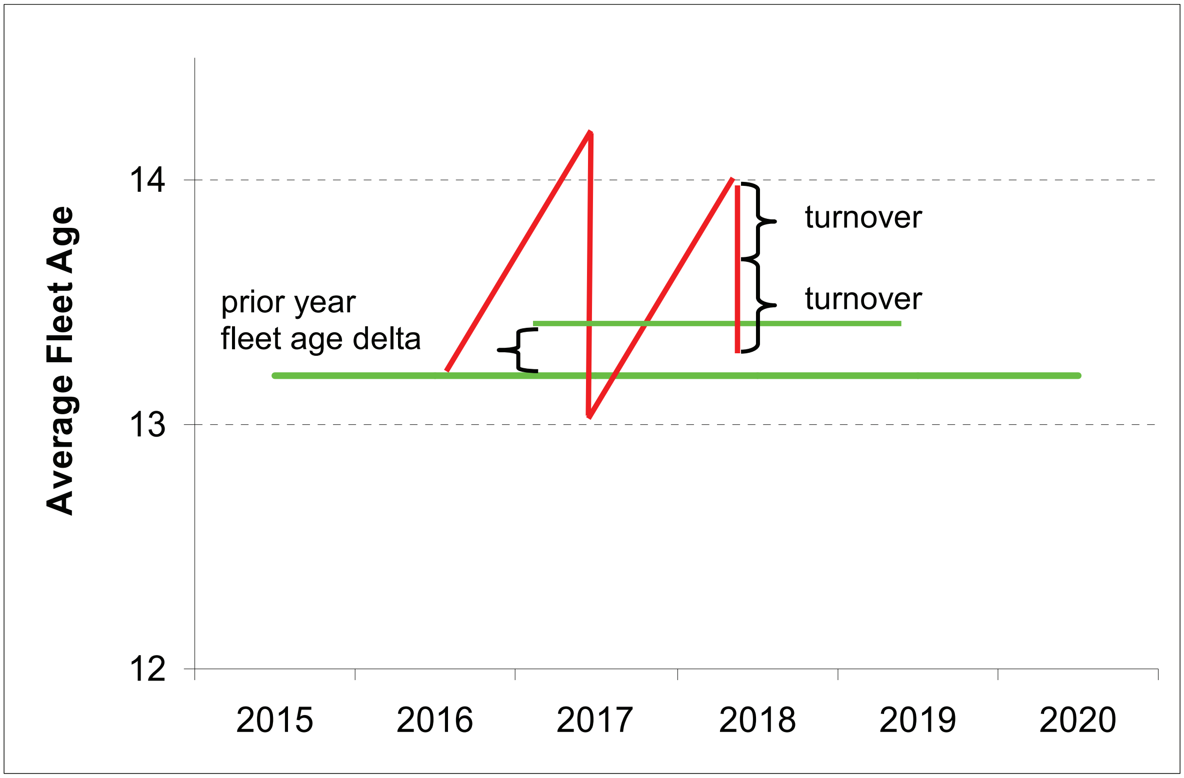


Figure 8: Fleet Age Delta

By incorporating the fleet age delta, fleet age will sometimes be less than the 2010 average fleet age and sometime more than the 2010 fleet average age, however over time the fleet average age will asymptotically approach the 2010 fleet average age. Fleets with a large number of vehicles will in general have a smaller fleet age delta than fleets with fewer vehicles. Small fleets will have a "lumpy" fleet average age over time exhibiting a periodic decrease and increase in fleet age. With a large number of fleets included in a simulation, the fluctuations of individual fleet ages will average out. Additional coding is required to shift the periodic behavior of fleets in 2010 to minimize cumulative impacts of all fleets that would otherwise all start the periodic decrease and increase in fleet average age at the same time.

It should also be noted that a vehicle is always replaced with a vehicle of the same vehicle type and same horsepower but a newer vehicle and engine model year. Fleets neither grow nor shrink in the OSM simulations since there is no basis for such in any given fleet. However, emissions are grown at a later step in the inventory process.

OSM excludes vehicles that have been recently retrofitted from the pool of vehicles subject to turnover; i.e., vehicles retrofit within the last six years. From the pool of eligible vehicles, OSM will rank them by relative age, where relative age is calculated using Equation 7. (For reproducibility between OSM runs, vehicles are also ranked by EIN as this provides a unique ordering of all vehicles.)

Equation 7: $$\mathrm{Re}\,lativeAge = \frac{CalendarYear - ModelYear}{AverageAge}$$

Where: Average age by vehicle type as shown in Table 2.

From the vehicle ranking, OSM will select the relative oldest vehicle for turnover. For example, a fleet may have a 10 year old skid steer and a 15 year old scraper. The skid steer has a relative age of 10/7.3 = 1.4 and the scraper has a relative age of 15/17.6 = 0.9 so the 10 year old skid steer will be turned over before the 15 year old scaper. No vehicles were modeled as repowers, which is a cost conservative estimate since repowers would be less costly than replacement.

OSM assumes that an older fleet would typically purchase newer but used vehicles to replace an older vehicle that they are selling or retiring and a younger fleet would purchase newer vehicles to replace a vehicle that they are selling or retiring. The fleet age is used to determine the typical age of a replacement vehicle. Also, OSM assumes that fleet owner business practices would change under the regulation, with fleet owners buying slightly newer vehicles than without the regulation.

Since different vehicle types have different average ages, a replacement vehicle age factor is used where:

Equation 8: $$\mathrm{Re}\,placementVehicleAge = \mathrm{Re}\,placementVehicleAgeFactor * VehicleAverageAge$$

Both the vehicle model year and the engine model year for the new, possibly used vehicle will be as shown in Equation 9.

Equation 9: $$\mathrm{Re}\,placementEngineModelYear = \mathrm{Re}\,gulatoryYear - \mathrm{Re}\,placementVehicleAge$$

The average age of the various vehicle types is shown in Table 2.

Table 2: Average Age by Vehicle Type

Vehicle Type	Average Age (Years)
A/C Tug Narrow Body	14.7
A/C Tug Wide Body	13.8
Aerial Lifts	5.9
Baggage Tug	20.5
Belt Loader	17.5
Bobtail	10.5
Bore/Drill Rigs	9.8
Cargo Loader	9.6
Cargo Tractor	15.3
Cranes	18.1
Crawler Tractors	16.4
Drill Rig (Mobile)	13.1
Excavators	9.2
Forklift (GSE)	17.2
Forklifts	11.5
Graders	17.9
Lift (GSE)	10.2
Off-Highway Tractors	12.0
Off-Highway Trucks	13.0
Other Construction Equipment	13.6
Other General Industrial Equipment	9.8
Other GSE	9.9
Other Material Handling Equipment	11.0
Passenger Stand	8.8
Pavers	11.1
Paving Equipment	11.4
Rollers	11.4
Rough Terrain Forklifts	8.1
Rubber Tired Dozers	21.6
Rubber Tired Loaders	13.6
Scrapers	17.6
Skid Steer Loaders	7.3
Surfacing Equipment	12.1
Sweepers/Scrubbers	11.1
Tractors/Loaders/Backhoes	10.9
Trenchers	12.1
Workover Rig (Mobile)	12.3

The default replacement factors are shown in Table 3.

Table 3: Replacement Factor By Fleet Age

Fleet age	Replacement Factor Baseline	Replacement Factor Scenario
< 8	0	0
8 to < 12	0.17	0.08
12 to < 16	0.33	0.17
16 to < 20	0.50	0.25
20 plus	0.75	0.50

In the base run OSM stops after doing the base turnover but in the scenario run OSM continues turnover. Since the base turnover occurs in the scenario run, the total horsepower actually turned over in the base turnover has to be included in the turnover carryover. Turnover carryover accounts for any incremental horsepower above that required by the regulation in a given year. Again, this occurs because discrete vehicles are turned over and the required turnover horsepower is often exceeded by some portion of the last engine horsepower turned over.

c. get_fleet_target.php, get_fleet_average.php, set_nox_pm_compliance_flags.php

To run the do_turnover.php page, several other pages need to be run first to set values used by the do_turnover.php page.

Using the vehicle model year and horsepower, OSM will look up the emissions targets as outlined in the rule and the associated emissions factors for the respective engines. OSM includes a retrofit factor which equals one unless the vehicle is retrofit, and then the retrofit factor is set to the percent of emissions that would result after retrofitting; e.g., a vehicle retrofit with a Level 3 device which reduces PM emissions by 85 percent would have a retrofit factor of 0.15. OSM will calculate the Fleet Targets and Fleet Average using Equation 8 and Equation 9 respectively.

Equation 10: $$FleetTarget = \frac{\sum Horsepower * EmissionTarget}{TotalHorsepower}$$

Equation 11: $$FleetAverage = \frac{\sum Horsepower * EmissionFactor * RetrofitFactor}{TotalHorsepower}$$

After OSM has calculated the fleet NOx and PM averages and targets, OSM will compare them and set compliance flags, i.e., it will set certain variables that will determine whether the turnover and retrofit pages are executed. If the NOx compliance flag is set to false, then OSM will attempt to turnover vehicles until the fleet does meet the NOx fleet target. If the PM compliance flag is set to false, then OSM will attempt to retrofit vehicles until the fleet does meet the PM fleet target.

OSM will do all turnover before it begins retrofitting, this is because turnover will reduce both NOx and PM, whereas retrofits (assuming retrofits do not control NOx) will only reduce PM thereby reducing the total retrofits and the associated costs to the fleet owner.

d. set_turnover_rate_and veh_pool, do_turnover.php

OSM assumes that a fleet will turn over vehicles under both the baseline and scenario runs, however unlike the base run which turns over to keep the fleet age constant, the additional turnover under the scenario run is in response to the regulatory requirements. Under the regulation a fleet must either meet the fleet average or meet a BACT requirement of turning over a certain percent of the total horsepower; this turnover is referred to as forced turnover. (A fleet may also run out of eligible vehicles before meeting the fleet average or BACT in which case it is compliant without further action for that regulatory year.) The forced turnover is calculated using Equation 12.

Equation 12 $ForcedTurnover = TotalHP * ForcedTurnoverRate$

Where ForcedTurnoverRate is the NOx BACT rate.

OSM excludes vehicles that are exempt under the regulation from the pool of vehicles subject to turnover; e.g., vehicles less than 10 years old, Tier 4 vehicles and before 2013, Tier 1 or newer vehicles. From the pool of eligible vehicles, whereas in the base run OSM will rank vehicles by relative age, in the scenario run OSM will rank them by Tier and then by relative age, where relative age is calculated, as in the base turnover, Equation 7.

Since the regulation provides exemptions and credits based on modifying the fleet's Tier composition, it is assumed that the fleets will preferentially replace older Tiered vehicles and then within that tier, the oldest by relative age. From the vehicle ranking, OSM will select the lowest tier and relative oldest vehicle for turnover. No vehicles were modeled as repowers, this is a cost conservative estimate since repowers would be less costly that a replacement.

The do_turnover.php page records into the osm_vehicle_scenario table the regulatory year the vehicle is turned over. Also, the do_turnover.php page inserts new tier, PM and NOx indexes into the osm_vehicle_scenario table looked up from zengine_index table, then recalculates the fleet NOx average and resets the compliance flags after each engine is turned over. If the fleet still does not meet the fleet NOx targets and there are still vehicles eligible for turnover, then the page repeats the turnover process until the fleet either meet the fleet NOx target or the fleet hits the maximum required turnover under BACT, or there are no more vehicles eligible for turnover or repower.

In instances where a fleet turns over more horsepower than is required, OSM will accumulate and track turnover carryover for the fleet. Carryover will accrue when the last vehicle required to be turned over does not match exactly the maximum horsepower required to be turned over, i.e., the 8 or 10 percent or when a fleet for other business purposes turns over more than required. The turnover carryover can be used in subsequent years to meet the maximum horsepower required to be turned over.

Any credit accrued due to early replacement, repowers, turnover, double NOx retrofit, and permanent low use, or reduced activity provisions are added to the turnover carryover in 2010. The reduced activity credits expire in 2012, the other credits continue until exhausted.

e. set_retrofit_rate_and veh_pool, do_retrofit.php

OSM assumes that fleets will not retrofit any vehicles under the baseline run nor when a fleet meets the PM target. Retrofits are assumed to be driven solely by the regulation; i.e., OSM assumes the fleet owner has no motivation to retrofit in the absence of the regulation. Hence, unlike the turnover rate, the retrofit rate is fixed at the PM BACT rate. OSM will retrofit eligible vehicles as long as a fleet does not meet the PM target or if it has not yet retrofit at least the percent of its fleet horsepower required under PM BACT.

OSM excludes from retrofit vehicles that are exempt under the regulation from the pool of vehicles subject to retrofit; i.e., vehicles less than five years old, tier 4s, and vehicles that have already been retrofit in the last ten years. After ten years OSM assumes that retrofitted vehicles will have the existing retrofit replaced with a similar device. This does not change the emissions associated with that vehicle but does add the cost of the new retrofit device in that regulatory year.

OSM utilizes the emissions controls associated with specific retrofit models for the data retrieved from DOORS. For example, if a vehicle has installed a Cleaire Longview DPF which is a Level 3 device with 25 percent NOx control, OSM will use a 0.75 NOx retrofit factor and a 0.15 PM retrofit factor when calculating the emissions associated with the vehicle. For all new retrofits installed under the scenario run the devices are assumed to be Level 3s that achieve at least 85% PM control and no NOx control.

Installing retrofits will change the fleet composition over time compared to no retrofits by essentially blocking retrofitted vehicles from turning over; i.e., vehicles have a six year exemption from turnover from the time a vehicle is first retrofit.

The do_retrofit.php page inserts into the osm_vehicle_scenario table a PM factor in a field in the osm_vehicle_scenario table which reduces the PM attributed to the vehicle and records the regulatory year the vehicle is retrofit. The do_retrofit.php page then recalculates the fleet PM average and resets the compliance flags after each engine is retrofit. If the fleet still does not meet the fleet PM targets and there are still vehicles eligible for retrofit, then the page repeats the retrofit process until the fleet either meet the fleet PM target, hits the maximum required retrofits, or there are no more vehicles eligible for retrofit.

Retrofit carryover, similar to turnover carryover, accumulates as the last vehicle to be retrofit exceeds the maximum required or due to business practices that retrofit more horsepower than is required. The retrofit carryover for medium and large fleets is typically negligible. On the other hand, retrofit carryover may be significant for small fleets, and indeed, the carryover provisions were developed to accommodate the needs of the small fleets.

For example, if a small fleet with just three older vehicles, all at 333 horsepower, did not meet the PM target, then in 2015 the fleet would be required to retrofit one vehicle representing approximately 33% of the fleet horsepower accruing 13% retrofit carryover in excess of the 20% required. If the fleet does not meet the PM fleet target in 2016 it will have to once again retrofit since the retrofit carryover not meet the required 20% maximum retrofit. After retrofitting a second engine in 2016, the fleet would have accrued 26% retrofit carryover credit which could be used in 2017 in lieu of meeting the PM fleet target. Although this example is not likely, it is provided to illustrate how OSM would utilize retrofit carryover.

f. write_fleet_summary.php, move_fleet_to_base_scenario.php

All of the preceding turnover and retrofit would be modeled for a given regulatory year on multiple vehicles for a specific fleet. All of the vehicle data would be recorded in the osm_vehicle table.

After OSM repeats the page for turnover until the fleet meets the NOx fleet average, or the fleet hits its maximum NOx BACT required, or there are no eligible engines OSM transfers control to the retrofit page. After OSM repeats the page for retrofit until the fleet meets the PM fleet average, or the fleet hits its maximum required PM BACT, or there are no eligible engines OSM transfer control to the write_fleet_summary.php page at the end of each year simulated.

The write_fleet_summary.php page records a snapshot of the entire fleet for a given regulatory year. OSM records for the regulatory year the final NOx and PM fleet averages, the percent turn over and retrofit, and any carryover. OSM will iterate through all of the regulatory years for a selected fleet through the selected end year up to 2030. Then, the move_fleet_to_base_scenario.php page will move all of the fleet data stored in osm_vehicle into either the osm_vehicle_base or osm_vehicle_scenario tables depending on whether it is a base or scenario run (the run value).

B. PHP Code

Note: The osm.php file is the calling file for all other osm1085 files and is therefore listed as the first in this document; all other php files are listed in alphabetical order; each file is separated from the next file by a line as shown here:

```
<?php  #osm

require_once ('connect.php');

include("doors_functions_1085.php");
include("fleet_stats_2010_1085.php");

$start_time = date("D M-d g:i A");
include ("initialize_run.php");

if (isset($_POST['run_sim'])
  or isset($_POST['run_sim1']))
{
    mysqli_query ($dbc4, "TRUNCATE TABLE osm_vehicle_base");
    mysqli_query ($dbc4, "TRUNCATE TABLE osm_vehicle_scenario");
    include ("run_sim.php");

    $query = 'TRUNCATE TABLE osm_fleet_base_costs_cumulative';
    $result = mysqli_query($dbc4, $query);

    $query = 'TRUNCATE TABLE osm_fleet_scenario_costs_cumulative';
    $result = mysqli_query($dbc4, $query);

    include ("costs.php");
    $message = "Costs Calculated.";
}

?>

<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">

<TABLE border="0" cellpadding="0" cellspacing="0" width="99%" bgcolor="#CEEFBD">

    <TR><TD><center><BR>

<?php
if (isset($_POST['run_sim'])
  or isset($_POST['run_sim1']))
{
 //   include ("show_costs_totals.php");
```

```
}
?>

   <TABLE BORDER="2" cellpadding="2" CELLSPACING="1" bgcolor="#D0D0D0">
      <TR><TD>

   <TABLE BORDER="0" cellpadding="2" CELLSPACING="1" bgcolor="#D0D0D0">
      <TR><TD colspan="4" bgcolor="#FFE7C6">
      <fieldset><CENTER><H2>Off-road Simulation Model (OSM)</H2>
      </TD></TR>
      <TR><TD colspan="2">

      <TABLE BORDER="0" cellpadding="2" CELLSPACING="1"  width="100%"
bgcolor="#FFFFC6">
      <TR><TD colspan="3"><BR><CENTER>
            <INPUT TYPE="SUBMIT" NAME="run_sim" VALUE="Run regulation
simulation"> <BR><BR>
            <INPUT TYPE="SUBMIT" NAME="run_sim1" VALUE="Run proposed
alternative"> <BR><BR>
<?php

   $color = "BLUE";
   if(isset($_SESSION['color']) and $_SESSION['color']=="RED") $color="BLUE";
   if(isset($_SESSION['color']) and $_SESSION['color']=="BLUE") $color="RED";
   $_SESSION['color']=$color;

   echo '<FONT COLOR="' . $color . '"><B>';
   echo $message;
   echo '</B></FONT><BR><BR><BR>';

   echo 'start - ' . $start_time . '<BR>'
        . 'end - ' . date("D M-d g:i A") . '<BR>';

?>
       </TD></TR></TABLE>
      </TD></TABLE></TABLE>
  <BR><BR> </TD></TR></TABLE>

</form>
```

```
<?php  # calculate_early_credit


$expiring_nox_credit_begin =  $reduced_activity_credit;
```

```
$turnover_carryover_begin  =  $retirement_2006_2010_credit
                        + $early_replacement_credit
                        + $repower_hp
                        + $turnover09_10
                        + $double_nox_retrofit_credit
                        + $permanent_low_use_nox_credit;
$free_2014 = false;

if(isset($_POST['run_sim1']))
{
   $turnover_carryover_begin  =  $repower_hp
                           + $double_nox_retrofit_credit
                           + $permanent_low_use_nox_credit;

   if($early_replacement_credit
      + $retirement_2006_2010_credit
        + $repower_hp
        + $turnover09_10
        > $nox_bact_hp)
   {
      $free_2014=true;
   }
}

$expiring_pm_credit_begin  =  $reduced_activity_credit
                        + $tier0_retirement_vdecs_credit;

$retrofit_carryover_begin  =  $retirement_2006_2010_credit
                        + $double_retrofit_credit
                        + $single_retrofit_credit;

$turnover_carryover = $turnover_carryover_begin;
$retrofit_carryover = $retrofit_carryover_begin;

$expiring_nox_credit = $expiring_nox_credit_begin;
$expiring_pm_credit  = $expiring_pm_credit_begin;

?>
```

```
<?php #connect

date_default_timezone_set('America/Los_Angeles');

DEFINE ('DB_HOST', 'localhost');
```

```
DEFINE ('DB_USER', 'puser');
DEFINE ('DB_PASSWORD', 'password');
DEFINE ('DB_NAME', 'doors1085');

$dbc = mysqli_connect (DB_HOST, DB_USER, DB_PASSWORD, DB_NAME)
                OR die(mysqli_connect_error());

DEFINE ('DB_HOST4', 'localhost');
DEFINE ('DB_USER4', 'puser');
DEFINE ('DB_PASSWORD4', 'password');
DEFINE ('DB_NAME4', 'osm1085');

$dbc4 = mysqli_connect (DB_HOST4, DB_USER4, DB_PASSWORD4, DB_NAME4)
         OR die(mysqli_connect_error());

?>
```

```
<?php # costs

$query = "UPDATE ztier4_premium_cost
        SET tier4_premium_cost = '$cost_tier4_premium_50'
        WHERE min_hp = 25
          AND max_hp = 50";
if(!$result = @mysqli_query($dbc4, $query)) echo 'updatet';

$query = "UPDATE ztier4_premium_cost
        SET tier4_premium_cost = '$cost_tier4_premium_175'
        WHERE min_hp = 50
          AND max_hp = 175";
if(!$result = @mysqli_query($dbc4, $query)) echo 'updatet';

$query = "UPDATE ztier4_premium_cost
        SET tier4_premium_cost = '$cost_tier4_premium_400'
        WHERE min_hp = 175
          AND max_hp = 400";
if(!$result = @mysqli_query($dbc4, $query)) echo 'updatet';

$query = "UPDATE ztier4_premium_cost
        SET tier4_premium_cost = '$cost_tier4_premium_400_plus'
        WHERE min_hp = 400
          AND max_hp = 9999";
if(!$result = @mysqli_query($dbc4, $query)) echo 'updatet';


$query = "UPDATE zretrofit_cost
```

```
        SET retrofit_cost = '$cost_retrofit_50'
        WHERE min_hp = 25
          AND max_hp = 50";
if(!$result = @mysqli_query($dbc4, $query)) echo 'updatet';

$query = "UPDATE zretrofit_cost
        SET retrofit_cost = '$cost_retrofit_125'
        WHERE min_hp = 50
          AND max_hp = 125";
if(!$result = @mysqli_query($dbc4, $query)) echo 'updatet';

$query = "UPDATE zretrofit_cost
        SET retrofit_cost = '$cost_retrofit_175'
        WHERE min_hp = 125
          AND max_hp = 175";
if(!$result = @mysqli_query($dbc4, $query)) echo 'updatet';

$query = "UPDATE zretrofit_cost
        SET retrofit_cost = '$cost_retrofit_300'
        WHERE min_hp = 175
          AND max_hp = 300";
if(!$result = @mysqli_query($dbc4, $query)) echo 'updatet';

$query = "UPDATE zretrofit_cost
        SET retrofit_cost = '$cost_retrofit_400'
        WHERE min_hp = 300
          AND max_hp = 400";
if(!$result = @mysqli_query($dbc4, $query)) echo 'updatet';

$query = "UPDATE zretrofit_cost
        SET retrofit_cost = '$cost_retrofit_600'
        WHERE min_hp = 400
          AND max_hp = 600";
if(!$result = @mysqli_query($dbc4, $query)) echo 'updatet';

$query = "UPDATE zretrofit_cost
        SET retrofit_cost = '$cost_retrofit_600_plus'
        WHERE min_hp = 600
          AND max_hp = 9999";
if(!$result = @mysqli_query($dbc4, $query)) echo 'updatet';

$query0 = "SELECT doors_id, count(*)
          FROM `osm_vehicle_base`
          WHERE `calendar_year` = 2010
          AND doors_id != 2369
          AND doors_id != 1972
```

```
          AND doors_id != 1560
          GROUP BY doors_id
          ORDER BY count(*), doors_id";
$result0 = @mysqli_query($dbc4, $query0);

while ($row0 = mysqli_fetch_assoc($result0))
{
   $doors_id = $row0['doors_id'];

   $repower_cost = 0;
   $price_new = 0;
   $price_life = 0;
   $price_old = 0;
   $turnover_cost = 0;
   $full_turnover_cost = 0;

   $query = "TRUNCATE TABLE osm_vehicle_base_costs";
   if(!$result = @mysqli_query($dbc4, $query)) echo 'drop';

   $query = "INSERT INTO osm_vehicle_base_costs
            (doors_id, ein, calendar_year, veh_type, veh_model_year,
             eng_model_year, eng_hp, hp_max, calendar_year_replace,
             calendar_year_retrofit, veh_average_age,
             eng_index_tier)

            SELECT
             doors_id, ein, calendar_year, veh_type, veh_model_year,
             eng_model_year, eng_hp, hp_max, calendar_year_replace,
             calendar_year_retrofit, veh_average_age,
             eng_index_tier

            FROM osm_vehicle_base
            WHERE doors_id = $doors_id";
   if(!$result = @mysqli_query($dbc4, $query)) echo 'insert';

   $query = "UPDATE osm_vehicle_base_costs
            INNER JOIN $vehicle_cost_file_name
              ON (osm_vehicle_base_costs.veh_type = $vehicle_cost_file_name.veh_type)
                AND (osm_vehicle_base_costs.calendar_year
                     - osm_vehicle_base_costs.veh_model_year =
$vehicle_cost_file_name.veh_age)
            INNER JOIN ztier4_premium_cost
              ON (osm_vehicle_base_costs.eng_hp > min_hp
              AND osm_vehicle_base_costs.eng_hp <= max_hp)
            SET osm_vehicle_base_costs.price_new =
            IF(eng_index_tier>'T3',
```

```
                ($vehicle_cost_file_name.veh_replacement_cost
                                           + tier4_premium_cost/eng_hp),
                ($vehicle_cost_file_name.veh_replacement_cost))
           WHERE osm_vehicle_base_costs.calendar_year =
osm_vehicle_base_costs.calendar_year_replace";
   if(!$result = @mysqli_query($dbc4, $query)) echo 'update1';

   $query = "UPDATE osm_vehicle_base_costs
           INNER JOIN zvehicle_cost
             ON (osm_vehicle_base_costs.veh_type = zvehicle_cost.veh_type)

           INNER JOIN osm_vehicle_base_costs as prior_year
             ON (osm_vehicle_base_costs.ein = prior_year.ein
                AND osm_vehicle_base_costs.calendar_year-1 =
prior_year.calendar_year)

           SET osm_vehicle_base_costs.price_old = veh_replacement_cost
           WHERE osm_vehicle_base_costs.calendar_year =
osm_vehicle_base_costs.calendar_year_replace
              AND osm_vehicle_base_costs.calendar_year - prior_year.veh_model_year
= veh_age";
   if(!$result = @mysqli_query($dbc4, $query)) echo 'update3';

   $query = "UPDATE osm_vehicle_base_costs
           SET  cost_full_turnover = (price_new + '$cost_transport' - price_old) * eng_hp
           WHERE calendar_year = calendar_year_replace";
   if(!$result = @mysqli_query($dbc4, $query)) echo 'update5';

   $query = "TRUNCATE TABLE osm_fleet_base_costs";
   $result = @mysqli_query($dbc4, $query);

   $query = "INSERT INTO osm_fleet_base_costs(doors_id, calendar_year)
                  (SELECT DISTINCT doors_id, calendar_year
                    FROM osm_vehicle_base_costs)";
   $result = @mysqli_query($dbc4, $query);

   $query = "SELECT doors_id, calendar_year,
                 SUM(cost_full_turnover) AS cost_annual_full_turnover,
                 SUM(cost_retrofit)      AS cost_annual_retrofit
           FROM osm_vehicle_base_costs
           GROUP BY doors_id, calendar_year";
   $result = @mysqli_query($dbc4, $query);

   while($row=mysqli_fetch_assoc($result))
   {
      $doors_id                 = $row['doors_id'];
```

```
        $calendar_year            = $row['calendar_year'];
        $cost_annual_full_turnover = $row['cost_annual_full_turnover'];
        $cost_annual_retrofit      = $row['cost_annual_retrofit'];

        if($cost_annual_full_turnover===NULL)  $cost_annual_full_turnover=0;
        if($cost_annual_retrofit===NULL)       $cost_annual_retrofit=0;

        $discount = pow((1+$discount_rate/100), -(integer)($calendar_year -
$discount_year));

        $cost_annual_full = $cost_annual_full_turnover
                      + $cost_annual_retrofit;

        $cost_annual_full_pv = $cost_annual_full * $discount;

     $cost_annual_full_turnover  = (integer)$cost_annual_full_turnover;
     $cost_annual_retrofit       = (integer)$cost_annual_retrofit;
     $cost_annual_full           = (integer)$cost_annual_full;
     $cost_annual_full_pv        = (integer)$cost_annual_full_pv;

        $query2 = "UPDATE osm_fleet_base_costs
              SET cost_annual_full_turnover  = '$cost_annual_full_turnover',
                  cost_annual_retrofit       = '$cost_annual_retrofit',
                  cost_annual_full           = '$cost_annual_full',
                  cost_annual_full_pv        = '$cost_annual_full_pv'
              WHERE doors_id = '$doors_id'
                AND calendar_year = '$calendar_year'";
        if(!$result2 = @mysqli_query($dbc4, $query2)) echo 'update6';
     }

     $query = "TRUNCATE TABLE osm_vehicle_scenario_costs";
     if(!$result = @mysqli_query($dbc4, $query)) echo 'drop';

     $query = "INSERT INTO osm_vehicle_scenario_costs
           (doors_id, ein, calendar_year, veh_type, veh_model_year,
            eng_model_year, eng_hp, hp_max, calendar_year_replace,
            calendar_year_retrofit, veh_average_age,
            eng_index_tier)

           SELECT
            doors_id, ein, calendar_year, veh_type, veh_model_year,
            eng_model_year, eng_hp, hp_max, calendar_year_replace,
            calendar_year_retrofit, veh_average_age,
            eng_index_tier

           FROM osm_vehicle_scenario
```

```
          WHERE doors_id = $doors_id";
   if(!$result = @mysqli_query($dbc4, $query)) echo 'insert';

   $query = "UPDATE osm_vehicle_scenario_costs
          INNER JOIN $vehicle_cost_file_name
            ON (osm_vehicle_scenario_costs.veh_type =
$vehicle_cost_file_name.veh_type)
              AND (osm_vehicle_scenario_costs.calendar_year
                   - osm_vehicle_scenario_costs.veh_model_year =
$vehicle_cost_file_name.veh_age)
          INNER JOIN ztier4_premium_cost
            ON (osm_vehicle_scenario_costs.eng_hp > min_hp
            AND osm_vehicle_scenario_costs.eng_hp <= max_hp)
          SET osm_vehicle_scenario_costs.price_new =
          IF(eng_index_tier>'T3',
              ($vehicle_cost_file_name.veh_replacement_cost
                                        + tier4_premium_cost/eng_hp),
              ($vehicle_cost_file_name.veh_replacement_cost))
          WHERE osm_vehicle_scenario_costs.calendar_year =
osm_vehicle_scenario_costs.calendar_year_replace";
   if(!$result = @mysqli_query($dbc4, $query)) echo 'update1';

   $query = "UPDATE osm_vehicle_scenario_costs
          INNER JOIN zvehicle_cost
            ON (osm_vehicle_scenario_costs.veh_type = zvehicle_cost.veh_type)

          INNER JOIN osm_vehicle_scenario_costs as prior_year
            ON (osm_vehicle_scenario_costs.ein = prior_year.ein
               AND osm_vehicle_scenario_costs.calendar_year-1 =
prior_year.calendar_year)

          SET osm_vehicle_scenario_costs.price_old = veh_replacement_cost
          WHERE osm_vehicle_scenario_costs.calendar_year =
osm_vehicle_scenario_costs.calendar_year_replace
              AND osm_vehicle_scenario_costs.calendar_year -
prior_year.veh_model_year = veh_age";
   if(!$result = @mysqli_query($dbc4, $query)) echo 'update3';

   $query = "UPDATE osm_vehicle_scenario_costs
          SET  cost_full_turnover = (price_new + '$cost_transport' - price_old) * eng_hp
          WHERE calendar_year = calendar_year_replace";
   if(!$result = @mysqli_query($dbc4, $query)) echo 'update5';

   $query = "UPDATE osm_vehicle_scenario_costs, zretrofit_cost
          SET  osm_vehicle_scenario_costs.cost_retrofit = zretrofit_cost.retrofit_cost
          WHERE calendar_year = calendar_year_retrofit
```

```
            AND osm_vehicle_scenario_costs.eng_hp > zretrofit_cost.min_hp
            AND osm_vehicle_scenario_costs.eng_hp < zretrofit_cost.max_hp";
    if(!$result = @mysqli_query($dbc4, $query)) echo 'update5';

    $query = "UPDATE osm_vehicle_scenario_costs, zretrofit_cost
            SET  osm_vehicle_scenario_costs.cost_retrofit = zretrofit_cost.retrofit_cost
            WHERE calendar_year = calendar_year_retrofit+10
              AND osm_vehicle_scenario_costs.eng_hp > zretrofit_cost.min_hp
              AND osm_vehicle_scenario_costs.eng_hp < zretrofit_cost.max_hp";
    if(!$result = @mysqli_query($dbc4, $query)) echo 'update5';

    $query = "TRUNCATE TABLE osm_fleet_scenario_costs";
    $result = @mysqli_query($dbc4, $query);

    $query = "INSERT INTO osm_fleet_scenario_costs(doors_id, calendar_year)
                    (SELECT DISTINCT doors_id, calendar_year
                     FROM osm_vehicle_scenario_costs)";
    $result = @mysqli_query($dbc4, $query);

    $query = "SELECT doors_id, calendar_year,
                  SUM(cost_full_turnover) AS cost_annual_full_turnover,
                  SUM(cost_retrofit)      AS cost_annual_retrofit
            FROM osm_vehicle_scenario_costs
            GROUP BY doors_id, calendar_year";
    $result = @mysqli_query($dbc4, $query);

    while($row=mysqli_fetch_assoc($result))
    {
       $doors_id                  = $row['doors_id'];
       $calendar_year             = $row['calendar_year'];
       $cost_annual_full_turnover = $row['cost_annual_full_turnover'];
       $cost_annual_retrofit      = $row['cost_annual_retrofit'];

       if($cost_annual_full_turnover===NULL)  $cost_annual_full_turnover=0;
       if($cost_annual_retrofit===NULL)       $cost_annual_retrofit=0;

       $discount = pow((1+$discount_rate/100), -(integer)($calendar_year -
$discount_year));

       $cost_annual_full = $cost_annual_full_turnover
                       + $cost_annual_retrofit;

       $cost_annual_full_pv = $cost_annual_full * $discount;

    $cost_annual_full_turnover  = (integer)$cost_annual_full_turnover;
    $cost_annual_retrofit       = (integer)$cost_annual_retrofit;
```

```
    $cost_annual_full          = (integer)$cost_annual_full;
    $cost_annual_full_pv       = (integer)$cost_annual_full_pv;

      $query2 = "UPDATE osm_fleet_scenario_costs
              SET cost_annual_full_turnover  = '$cost_annual_full_turnover',
                cost_annual_retrofit       = '$cost_annual_retrofit',
                cost_annual_full           = '$cost_annual_full',
                cost_annual_full_pv        = '$cost_annual_full_pv'
              WHERE doors_id = '$doors_id'
                AND calendar_year = '$calendar_year'";
      if(!$result2 = @mysqli_query($dbc4, $query2)) echo 'update6';
    }

    $query = 'REPLACE INTO osm_fleet_base_costs_cumulative
          SELECT *
          FROM osm_fleet_base_costs';
    $result = mysqli_query($dbc4, $query);

    $query = 'REPLACE INTO osm_fleet_scenario_costs_cumulative
          SELECT *
          FROM osm_fleet_scenario_costs';
    $result = mysqli_query($dbc4, $query);
}

?>
```

```
<?php  #create_fleet

    $doors_id_arr = array();
    $_SESSION['ren_name'] = 'zzz';
    $doors_id_arr[(integer)$doors_id] = $_SESSION['ren_name'];
    $_SESSION['doors_id_arr'] = $doors_id_arr;

    get_fleet_stats();

    $query = "SELECT *
          FROM temp_2010
          WHERE date_veh_sale IS NULL
            AND eng_hp >= 25";
    $result = @mysqli_query ($dbc, $query);

    while ($row = mysqli_fetch_assoc($result))
    {
      $ren_size       = $row['ren_size'];
      $ein            = $row['ein'];
```

```
    $veh_type       = $row['veh_type'];
    $veh_model_year = $row['veh_model_year'];
    $eng_model_year = $row['eng_model_year'];
    $eng_hp         = $row['eng_hp'];
    $vdecs_pm       = $row['vdecs_pm'];
    $vdecs_nox      = $row['vdecs_nox'];
    $low_use        = $row['low_use'];

    if (isset($row['date_vdecs_purchase']))
    {
      $date_vdecs_purchase = $row['date_vdecs_purchase'];
      $calendar_year_retrofit = substr($date_vdecs_purchase, 0, 4);
    }
    else
    {
      $calendar_year_retrofit = '0000';
    }

    $query4 = "INSERT INTO osm_vehicle (ren_size,
                doors_id, calendar_year, ein, veh_type,
                veh_model_year, eng_model_year, eng_hp, low_use,
                calendar_year_retrofit, vdecs_pm, vdecs_nox)
           VALUES ('$ren_size',
                '$doors_id', '2010', '$ein', '$veh_type',
                '$veh_model_year', '$eng_model_year', '$eng_hp', '$low_use',
                '$calendar_year_retrofit', '$vdecs_pm', '$vdecs_nox')";
              $result4 = @mysqli_query ($dbc4, $query4);
}

$query4 = "UPDATE osm_vehicle
        SET osm_vehicle.hp_max = 50
        WHERE osm_vehicle.eng_hp  >= 25
          AND osm_vehicle.eng_hp  <= 50";
              $result4 = @mysqli_query ($dbc4, $query4);

$query4 = "UPDATE osm_vehicle
        SET osm_vehicle.hp_max = 120
        WHERE osm_vehicle.eng_hp  > 50
          AND osm_vehicle.eng_hp  <= 120";
              $result4 = @mysqli_query ($dbc4, $query4);

$query4 = "UPDATE osm_vehicle
        SET osm_vehicle.hp_max = 175
        WHERE osm_vehicle.eng_hp  > 120
          AND osm_vehicle.eng_hp  <= 175";
              $result4 = @mysqli_query ($dbc4, $query4);
```

```
   $query4 = "UPDATE osm_vehicle
           SET osm_vehicle.hp_max = 250
           WHERE osm_vehicle.eng_hp  > 175
             AND osm_vehicle.eng_hp  <= 250";
                 $result4 = @mysqli_query ($dbc4, $query4);

   $query4 = "UPDATE osm_vehicle
           SET osm_vehicle.hp_max = 500
           WHERE osm_vehicle.eng_hp  > 250
             AND osm_vehicle.eng_hp  <= 500";
                 $result4 = @mysqli_query ($dbc4, $query4);

   $query4 = "UPDATE osm_vehicle
           SET osm_vehicle.hp_max = 750
           WHERE osm_vehicle.eng_hp  > 500
             AND osm_vehicle.eng_hp  <= 750";
                 $result4 = @mysqli_query ($dbc4, $query4);

   $query4 = "UPDATE osm_vehicle
           SET osm_vehicle.hp_max = 1000
           WHERE osm_vehicle.eng_hp  > 750
             AND osm_vehicle.eng_hp  <= 1000";
                 $result4 = @mysqli_query ($dbc4, $query4);

   $query4 = "UPDATE osm_vehicle
           SET osm_vehicle.hp_max = 9999
           WHERE osm_vehicle.eng_hp  > 1000
             AND osm_vehicle.eng_hp  <= 9999";
                 $result4 = @mysqli_query ($dbc4, $query4);

?>
```

```
<?php  #create_new_calendar_year

$calendar_year_minus1 = $calendar_year-1;

$query = "SELECT *
        FROM osm_vehicle
        WHERE calendar_year = $calendar_year_minus1";
$result= @mysqli_query($dbc4, $query);

while($row = mysqli_fetch_assoc($result))
{
   $ren_size              = $row['ren_size'];
```

```
    $ein                  = $row['ein'];
    $veh_type             = $row['veh_type'];
    $veh_model_year       = $row['veh_model_year'];
    $eng_model_year       = $row['eng_model_year'];
    $eng_hp               = $row['eng_hp'];
    $hp_max               = $row['hp_max'];
    $low_use              = $row['low_use'];
    $calendar_year_replace  = $row['calendar_year_replace'];
    $calendar_year_retrofit = $row['calendar_year_retrofit'];
    $vdecs_nox            = $row['vdecs_nox'];
    $vdecs_pm             = $row['vdecs_pm'];
    $veh_average_age      = $row['veh_average_age'];
    $veh_relative_age     = ($calendar_year - $veh_model_year) /$veh_average_age;
    $eng_index_tier       = $row['eng_index_tier'];
    $eng_index_nox        = $row['eng_index_nox'];
    $eng_index_pm         = $row['eng_index_pm'];


    $query2= "INSERT INTO osm_vehicle (ren_size,
             doors_id, calendar_year, ein, veh_type,
             veh_model_year, eng_model_year, eng_hp, hp_max, low_use,
             calendar_year_replace,
             calendar_year_retrofit, vdecs_nox, vdecs_pm,
             veh_average_age, veh_relative_age, eng_index_tier,
             eng_index_nox, eng_index_pm)
         VALUES ('$ren_size',
             '$doors_id', '$calendar_year', '$ein', '$veh_type',
             '$veh_model_year', '$eng_model_year', '$eng_hp', '$hp_max', '$low_use',
             '$calendar_year_replace',
             '$calendar_year_retrofit', '$vdecs_nox', '$vdecs_pm',
             '$veh_average_age', '$veh_relative_age', '$eng_index_tier',
             '$eng_index_nox', '$eng_index_pm')";

              $result2 = @mysqli_query ($dbc4, $query2);
}

include("insert_fleet_averages_targets.php");

?>
```

```
<?php  #do_retrofit

while($row = mysqli_fetch_assoc($result)
      and !$meets_fleet_target_pm
      and $retrofit_actual < $retrofit_needed)
```

```
{
   if ($calendar_year<=$end_year_double_retrofit)
   {
      $retrofit_actual += $row['eng_hp']*2;
   }
   else
   {
      $retrofit_actual += $row['eng_hp'];
   }
   $ein = $row['ein'];

   $query2 = "UPDATE osm_vehicle
           SET calendar_year_retrofit  = '$calendar_year',
              vdecs_pm      = 0.15
            WHERE ((calendar_year = '$calendar_year')
              AND (doors_id = '$doors_id')
              AND (ein = '$ein'))";
                 if(!$result2 = @mysqli_query ($dbc4, $query2)) $message .= '_264';

   include("get_fleet_average.php");
   include("set_nox_pm_compliance_flags.php");
}

if(!$row = mysqli_fetch_assoc($result))
{
   $retrofit_pool_empty = 'Yes';
}

if($retrofit_actual>0)
{
   if($meets_fleet_target_nox)
   {
      $retrofit_carryover = $retrofit_carryover_begin
                       +max($retrofit_actual
                            + $turnover_tier4
                            - $retrofit_rate_forced*$total_hp , 0);
   }
   else
   {
      $retrofit_carryover = max($retrofit_carryover +
                            $retrofit_actual - $retrofit_needed, 0);
   }

   $retrofit_pct = $retrofit_actual / $total_hp * 100;
}
```

```
?>
```

```
<?php  #do_turnover

if(!$result = @mysqli_query ($dbc4, $query_fleet)) $message .= '_59';

while($row = mysqli_fetch_assoc($result)
     and (!$meets_fleet_target_nox
          and $turnover_actual < $turnover_needed
          or ($turnover_only
             and !$meets_fleet_target_pm
             and $turnover_actual < $retrofit_needed)))
{
   $ein = $row['ein'];
   $veh_type = $row['veh_type'];

   $query4 = "SELECT ROUND($replacement_vehicle_age_factor * veh_average_age)
as replacement_vehicle_age
            FROM zvehicle_average_age
           WHERE veh_type = '$veh_type'";
        $result4 = @mysqli_query ($dbc4, $query4);

   $row4 = mysqli_fetch_assoc($result4);
   $replacement_vehicle_age= $row4['replacement_vehicle_age'];

   $model_year = $calendar_year-$replacement_vehicle_age;

   if($run==1)
   {
      $model_year = max($model_year, 1999);
   }

   $replacement_model_age = $calendar_year-$model_year;

   $query2 = "UPDATE osm_vehicle
           SET calendar_year_replace   = '$calendar_year',
              veh_model_year          = '$model_year',
              eng_model_year          = '$model_year',
              veh_relative_age        = '$replacement_model_age'/veh_average_age,
              vdecs_pm                = '1'
            WHERE  doors_id        = '$doors_id'
              AND calendar_year  = '$calendar_year'
              AND ein            = '$ein'";
                 if(!$result2 = @mysqli_query ($dbc4, $query2)) $message .= '.';
```

```
   include("insert_eng_index_target.php");

   include("get_fleet_average.php");
   include("set_nox_pm_compliance_flags.php");

   $turnover_actual += $row['eng_hp'];
}

if(!$row = mysqli_fetch_assoc($result))
{
   $turnover_pool_empty = 'Yes';
}

if($turnover_actual >0)
{
   if($meets_fleet_target_nox)
   {
      $turnover_carryover = $turnover_carryover_begin
                           + max($turnover_actual + $turnover_actual_base
                                - $turnover_rate_forced*$total_hp, 0);
   }
   else
   {
      $turnover_carryover = max($turnover_carryover_begin
                              + $turnover_actual + $turnover_actual_base
                              - $turnover_rate_forced*$total_hp, 0);
   }

   $turnover_pct = ($turnover_actual + $turnover_actual_base)/ $total_hp;
}


?>
```

```
<?php  #do_turnover_base

$cy_year = 'cy_' . $calendar_year;

if($economic_recovery=='na')
{
   $fleet_age_adjusted=$fleet_age_2010;
}
else
{
   $query0 = "SELECT SUM(eng_hp*(calendar_year-
```

```
                        IF(eng_model_year=2011 AND calendar_year=2010,
                                    2010,eng_model_year))
                                    * $cy_year)
                               / SUM(eng_hp)
                          AS fleet_age_adjusted
            FROM osm_vehicle
            INNER JOIN zequipment_class
               USING (veh_type)
            INNER JOIN zeconomic_recovery_factor
               USING (equipment_class)
            WHERE doors_id = $doors_id
              AND calendar_year = 2010
              AND economic_recovery = '$economic_recovery'
              AND low_use = 0";

   $result0 = @mysqli_query ($dbc4, $query0);
   $row0 = mysqli_fetch_assoc($result0);

   $fleet_age_adjusted = $row0['fleet_age_adjusted'];
}

$query_fleet = "SELECT ein, eng_hp, veh_type, low_use
              FROM osm_vehicle
              WHERE doors_id = '$doors_id'
                AND calendar_year = '$calendar_year'
                AND '$calendar_year' - calendar_year_retrofit >= 6
                AND low_use = 0
              ORDER BY veh_relative_age DESC, eng_index_tier, ein";

if(!$result = @mysqli_query ($dbc4, $query_fleet)) $message .= '_59';

while($row = mysqli_fetch_assoc($result)
     and ($fleet_age > $fleet_age_adjusted + $fleet_age_delta))
{
   $turnover_actual_base += $row['eng_hp'];

   $ein = $row['ein'];
   $veh_type = $row['veh_type'];
   $query4 = "SELECT ROUND($replacement_vehicle_age_factor * veh_average_age)
as replacement_vehicle_age
             FROM zvehicle_average_age
            WHERE veh_type = '$veh_type'";
        $result4 = @mysqli_query ($dbc4, $query4);

   $row4 = mysqli_fetch_assoc($result4);
   $replacement_vehicle_age= $row4['replacement_vehicle_age'];
```

```
   $model_year = $calendar_year-$replacement_vehicle_age;

   if($run==1)
   {
      $model_year = max($model_year, 1999);
   }

   $replacement_model_age = $calendar_year-$model_year;

   $query2 = "UPDATE osm_vehicle
            SET calendar_year_replace   = '$calendar_year',
               veh_model_year         = '$model_year',
               eng_model_year         = '$model_year',
               low_use                = '0',
               veh_relative_age       = '$replacement_model_age'/veh_average_age,
               vdecs_pm               = '1'
             WHERE  doors_id       = '$doors_id'
               AND calendar_year  = '$calendar_year'
               AND ein           = '$ein'";
                  if(!$result2 = @mysqli_query ($dbc4, $query2)) $message .= '
do_turnover_base line 72 ';

   include("insert_eng_index_target.php");

   include("get_fleet_info.php");
}

if(!$row = mysqli_fetch_assoc($result))
{
   $turnover_pool_empty = 'Yes';
}


if($turnover_actual_base >0)
{
   if($meets_fleet_target_nox)
   {
      $turnover_carryover = $turnover_carryover_begin
                         + max($turnover_actual_base
                              - $turnover_rate_forced*$total_hp, 0);

   }
   else
   {
      $turnover_carryover = max($turnover_carryover_begin
```

```
                        + $turnover_actual_base
                        - $turnover_rate_forced*$total_hp, 0);
    }

    $turnover_pct = $turnover_actual_base/ $total_hp;
}


if($run == 0
  or ($run == 1 and ($fleet_target_nox == 0
             or $meets_fleet_target_nox
             or $turnover_rate_forced == 0)))
{
    if($economic_recovery=='na')
    {
        $fleet_age_delta +=  $fleet_age_2010 - $fleet_age;
    }
    else
    {
        $fleet_age_delta +=  $fleet_age_adjusted - $fleet_age;
    }
}

?>
```

```
<?php #doors_functions_1085

//============================================================
============
function missing_engine()
{
GLOBAL $doors_id, $dbc, $message;

    $message = '';

    if(!isset($_SESSION['ren_name'])) $_SESSION['ren_name'] = 'missing_engine';

    if(!isset($_SESSION['doors_id_arr']))
    {
            $doors_id_arr = array();
            $doors_id_arr[(integer)$doors_id] = 'ren_name';
            $_SESSION['doors_id_arr'] = $doors_id_arr;
    }

    $doors_id_arr = $_SESSION['doors_id_arr'];
```

```
   $doors_id_key_arr = implode(", ", array_keys($doors_id_arr));


   $query = "SELECT ein, doors_id
            FROM rentity_vehicle
            LEFT JOIN vehicle
               USING (ein)
            LEFT JOIN zvehicle_man_mod
               USING (veh_man_mod_id)
            LEFT JOIN vehicle_engine
               USING (ein)
            LEFT JOIN engine
               USING (engine_id)
            LEFT JOIN engine_hp
               USING (engine_id)
            WHERE doors_id IN ($doors_id_key_arr)

              AND (engine.engine_id IS NULL
                 OR engine_hp.eng_hp IS NULL)";
   $result = @mysqli_query ($dbc, $query);

   if(mysqli_num_rows($result)>0)
   {
      while ($row = mysqli_fetch_assoc($result))
      {
         $message .= '<BR>Fleet DOORS ID: <FONT COLOR="BLACK">' .
$row['doors_id']
                 . '</FONT>, Vehicle with EIN: <FONT COLOR="BLACK">' . $row['ein']
                . " </FONT>does not have an engine."
                . " <BR>You must enter the engine"
                . " information before proceeding.<BR>";
      }
    }

    // need to trigger for any of the multiple doors_id
   if ($message !== '')
   {
      return true;
   }
   else
   {
      return false;
   }
}
```

```
//=====================================================================
=========
//

function create_temp_2010_table()
{
GLOBAL $doors_id, $dbc, $dbc2;

//=====================================================================
=========
// Capture all vehicle history for fleet and store in a temp table for all
// subsequent queries

   $query = "DROP TABLE `temp_2010`";
   if($result = @mysqli_query ($dbc, $query)); // echo 'Dropped temp table<BR>';

   // $doors_id_arr is necessary to aggregate fleets
   $doors_id_arr = $_SESSION['doors_id_arr'];
   $doors_id_key_arr = implode(", ", array_keys($doors_id_arr));

   $query = "CREATE TABLE temp_2010
         (PRIMARY KEY (doors_id, ein, engine_id, date_veh_purchase,
                                    date_eng_purchase, eng_hp))
         SELECT rentity_vehicle.doors_id, ren_size, vehicle.ein,
            date_veh_purchase, date_veh_inservice, date_veh_sale,
            veh_owner_id, veh_type, veh_manufacturer, veh_model,
            veh_model_year, veh_serial_number,
            NOT isnull(vehicle_low_use.ein) as low_use,
            NOT isnull(vehicle_low_use.ein) as repower,
            NOT isnull(vehicle_specialty.ein) as specialty,
            NOT isnull(engine_non_diesel.engine_id) as alt_fuel,
            NOT isnull(vehicle_program.ein) as program,
            engine_id, date_eng_purchase, date_eng_installed,
            date_eng_sale,
            eng_manufacturer, eng_model, eng_family, eng_serial_number,
            eng_model_year, eng_hp, eng_displacement,
            vdecs_id, vdecs_serial_number, date_vdecs_purchase,
            date_vdecs_sale, vdecs_family_name, 'vdecs_pm', 'vdecs_nox'
         FROM rentity_vehicle
         LEFT JOIN rentity_statistics
            USING (doors_id)
         LEFT JOIN vehicle
            USING (ein)
         LEFT JOIN vehicle_low_use
            USING (ein)
         LEFT JOIN vehicle_emergency
```

```
            USING (ein)
          LEFT JOIN vehicle_specialty
            USING (ein)
          LEFT JOIN vehicle_snow_removal
            USING (ein)
          LEFT JOIN vehicle_ag
            USING (ein)
          LEFT JOIN vehicle_awaiting_sale
            USING (ein)
          LEFT JOIN vehicle_island
            USING (ein)
          LEFT JOIN vehicle_program
            USING (ein)
          LEFT JOIN vehicle_engine
            USING (ein)
          LEFT JOIN zvehicle_man_mod
            USING (veh_man_mod_id)
          LEFT JOIN engine
            USING (engine_id)
          LEFT JOIN engine_hp
            USING (engine_id)
          LEFT JOIN engine_non_diesel
            USING (engine_id)
          LEFT JOIN zengine_man_mod
            USING (eng_man_mod_id)
          LEFT JOIN engine_vdecs
            USING (engine_id)
          LEFT JOIN vdecs
            USING (vdecs_id)
          WHERE doors_id IN ($doors_id_key_arr)
           AND vehicle_emergency.ein IS NULL
           AND vehicle_ag.ein IS NULL
           AND vehicle_snow_removal.ein IS NULL
           AND vehicle_awaiting_sale.ein IS NULL
           AND vehicle_island.ein IS NULL
           AND eng_hp >=25";
  if(!$result = @mysqli_query ($dbc, $query)) echo "_temp table didn't write. Two-
engine vehicles?";

$result = @mysqli_query($dbc, "Select * from temp_2010");

  $query = "UPDATE temp_2010
          SET vdecs_pm = '1',
            vdecs_nox = '1'";
  $result = @mysqli_query($dbc, $query);
```

```
   $query = "UPDATE temp_2010
            SET vdecs_pm = '0.15'
            WHERE vdecs_family_name IS NOT NULL";
   $result = @mysqli_query($dbc, $query);

   return;
}


?>
```

```
<?php #fleets_stats_2010

//==================================================================
========
function get_fleet_stats()
{
GLOBAL $doors_id,
       $ren_red_act_pct, $ren_red_act_2007, $ren_red_act_2010,
      $ren_red_act_method, $retirement_2006_2010_credit,
       $average_2007_hp_hrs, $activity_2007, $average_2007_hp, $activity_2010,
       $july_2007_hp, $march_2010_hp, $reduced_activity_credit,
       $total_hp_with_low_use , $total_hp_exempt, $num_veh_exempt,
        $num_veh_with_low_use,

      $total_hp_fleet_average_year_2006, $num_veh_fleet_average_year_2006,
      $total_hp_fleet_average_year_2007, $num_veh_fleet_average_year_2007,
      $total_hp_fleet_average_year_2008, $num_veh_fleet_average_year_2008,
      $total_hp_fleet_average_year_2009, $num_veh_fleet_average_year_2009,
      $total_hp_fleet_average_year_2010, $num_veh_fleet_average_year_2010,
      $total_hp_fleet_average_year_2011, $num_veh_fleet_average_year_2011,

      $num_veh_sold_2006_2011,
      $num_veh_added_2006_2009,
      $num_veh_added_2009_2010,
      $num_veh_added_2010_2011,

      $total_hp_low_use_year_2006, $num_veh_low_use_year_2006,
      $total_hp_low_use_year_2007, $num_veh_low_use_year_2007,
      $total_hp_low_use_year_2008, $num_veh_low_use_year_2008,
      $total_hp_low_use_year_2009, $num_veh_low_use_year_2009,
      $total_hp_low_use_year_2010, $num_veh_low_use_year_2010,
      $total_hp_low_use_year_2011, $num_veh_low_use_year_2011,
```

```
  $total_hp_low_use, $num_veh_low_use,
  $total_hp_repower, $num_veh_repower,
  $total_hp_retrofit, $num_veh_retrofit,
  $total_hp_alt_fuel, $num_veh_alt_fuel,
  $total_hp_program, $num_veh_program,
  $total_hp_t0, $num_veh_t0,
  $total_hp_t1U, $num_veh_t1U,
  $total_hp_t1Z, $num_veh_t1Z,
  $total_hp_t2, $num_veh_t2,
  $total_hp_t3, $num_veh_t3,
  $total_hp_t4I, $num_veh_t4I,

  $fleet_veh_age_by_count, $fleet_veh_age_by_hp, $fleet_eng_age_by_count,
  $fleet_eng_age_by_hp,  $early_replacement_credit,
  $hp_reduction_09_10, $lowest_tier_hp_reduction_09_10,
  $tier0_hp_reduction_09_10, $tier0_retirement_vdecs_credit,
  $repower_hp, $double_retrofit_credit, $single_retrofit_credit,
  $nox_credit, $pm_credit, $nox_bact_hp, $pm_bact_hp,
  $nox_fleet_average_2010, $pm_fleet_average_2010,


 $nox_fleet_target_year,
 $pm_fleet_target_year,

 $nox_fleet_target_2010,
 $nox_fleet_target_2011,
 $nox_fleet_target_2012,
 $nox_fleet_target_2013,
 $nox_fleet_target_2014,
 $nox_fleet_target_2015,
 $nox_fleet_target_2016,
 $nox_fleet_target_2017,
 $nox_fleet_target_2018,
 $nox_fleet_target_2019,
 $nox_fleet_target_2020,

 $pm_fleet_target_2010,
 $pm_fleet_target_2011,
 $pm_fleet_target_2012,
 $pm_fleet_target_2013,
 $pm_fleet_target_2014,
 $pm_fleet_target_2015,
 $pm_fleet_target_2016,
 $pm_fleet_target_2017,
 $pm_fleet_target_2018,
 $pm_fleet_target_2019,
```

```
$pm_fleet_target_2020,

  $meets_nox, $meets_pm, $ren_size,
  $meets_nox_fleet_average, $meets_nox_bact, $meets_pm_fleet_average,
  $meets_pm_bact, $turnover09_10, $double_nox_retrofit_credit,
  $permanent_low_use_nox_credit, $dbc, $message;

$ren_red_act_pct              = 0;
$ren_red_act_2007              = 0;
$ren_red_act_2010              = 0;
$ren_red_act_method             = 'NA';

$retirement_2006_2010_credit     = 0;

$average_2007_hp_hrs            = 0;
$activity_2007                 = 0;
$average_2007_hp               = 0;
$activity_2010                 = 0;
$july_2007_hp                  = 0;
$march_2010_hp                 = 0;
$reduced_activity_credit        = 0;

$total_hp_fleet_average_year_2006 = 0;
$total_hp_fleet_average_year_2007 = 0;
$total_hp_fleet_average_year_2008 = 0;
$total_hp_fleet_average_year_2009 = 0;
$total_hp_fleet_average_year_2010 = 0;
$early_replacement_credit       = 0;

$hp_reduction_09_10             = 0; // THIS should be = to 10 - 09
$lowest_tier_hp_reduction_09_10   = 0;
$tier0_hp_reduction_09_10        = 0;
$tier0_retirement_vdecs_credit    = 0;

$repower_hp                    = 0;
$double_retrofit_credit          = 0;
$single_retrofit_credit          = 0;

$nox_credit                    = 0;
$pm_credit                     = 0;
$nox_bact_hp                   = 0;
$pm_bact_hp                    = 0;

$nox_fleet_average_2010          = 0;
```

```
    $pm_fleet_average_2010           = 0;

    $nox_fleet_target_year          = 0;
    $pm_fleet_target_year           = 0;

    $nox_fleet_target_2010          = 0;
    $nox_fleet_target_2011          = 0;
    $nox_fleet_target_2012          = 0;
    $nox_fleet_target_2013          = 0;
    $nox_fleet_target_2014          = 0;
    $nox_fleet_target_2015          = 0;
    $nox_fleet_target_2016          = 0;
    $nox_fleet_target_2017          = 0;
    $nox_fleet_target_2018          = 0;
    $nox_fleet_target_2020          = 0;

    $pm_fleet_target_2010           = 0;
    $pm_fleet_target_2011           = 0;
    $pm_fleet_target_2012           = 0;
    $pm_fleet_target_2013           = 0;
    $pm_fleet_target_2014           = 0;
    $pm_fleet_target_2015           = 0;
    $pm_fleet_target_2016           = 0;
    $pm_fleet_target_2017           = 0;
    $pm_fleet_target_2018           = 0;
    $pm_fleet_target_2019           = 0;
    $pm_fleet_target_2020           = 0;

//==================================================================
=========

    if(isset($_SESSION['doors_id_arr'])
       and count($_SESSION['doors_id_arr'])>1)
    {
       $ren_size = 'L';
    }
    else
    {
       $calendar_year = "2010";
       $query = "SELECT ren_size
                 FROM rentity_statistics
                 WHERE doors_id = '$doors_id'
                   AND calendar_year = '$calendar_year'";
       $result = @mysqli_query($dbc, $query);
       $row = mysqli_fetch_assoc($result);
       $ren_size = $row['ren_size'];
```

```
   }

   create_temp_2010_table();


//======================================================================
===========
// Current fleet including low use

   $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                    COUNT(ein) as num_veh
                  FROM temp_2010
                  WHERE date_veh_sale IS NULL
                    AND date_eng_sale IS NULL";

   $result = @mysqli_query ($dbc, $query);
   $row = mysqli_fetch_assoc($result);
   $total_hp_with_low_use = $row['total_hp'];
   $num_veh_with_low_use = $row['num_veh'];

//======================================================================
=========
// vehicles (hp) in fleet average 2006-2010

for ($year_query=2006 ; $year_query<2012; $year_query++ )
{
   //  this creates a variable variable with names:
   //  num_veh_fleet_average_year_2006 and total_hp_fleet_average_year_2006
   //  num_veh_fleet_average_year_2007 and total_hp_fleet_average_year_2007
   //  etc. through 2011

   $num_veh_fleet_average_year_var = 'num_veh_fleet_average_year_' . $year_query;
   $total_hp_fleet_average_year_var = 'total_hp_fleet_average_year_' . $year_query;

   $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0)  as total_hp,
                    COUNT(ein) as num_veh
         FROM temp_2010
         WHERE !low_use
           AND date_veh_purchase < '$year_query-03-02'
           AND ((date_veh_sale IS NULL)
             OR (date_veh_sale > '$year_query-03-01'))
           AND date_eng_installed < '$year_query-03-02'
           AND (date_eng_sale IS NULL
             OR (date_eng_sale > '$year_query-03-01'))";

   $result = @mysqli_query ($dbc, $query);
```

```
    $row = mysqli_fetch_assoc($result);
    $$num_veh_fleet_average_year_var = $row['num_veh'];
    $$total_hp_fleet_average_year_var = $row['total_hp'];
}

//====================================================================
=========
// vehicles (hp) in low-use 2006-2010

for ($year_query=2006 ; $year_query<2012; $year_query++ )
{
    //  this creates a variable variable with names:
    //  num_veh_fleet_average_year_2006 and total_hp_fleet_average_year_2006
    //  num_veh_fleet_average_year_2007 and total_hp_fleet_average_year_2007
    //  etc. through 2011

    $num_veh_low_use_year_var = 'num_veh_low_use_year_' . $year_query;
    $total_hp_low_use_year_var = 'total_hp_low_use_year_' . $year_query;

    $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0)  as total_hp,
                        COUNT(ein) as num_veh
            FROM temp_2010
            WHERE low_use
              AND date_veh_purchase < '$year_query-03-02'
              AND ((date_veh_sale IS NULL)
                OR (date_veh_sale > '$year_query-03-01'))
              AND date_eng_installed < '$year_query-03-02'
              AND (date_eng_sale IS NULL
                OR (date_eng_sale > '$year_query-03-01'))";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
    $$num_veh_low_use_year_var = $row['num_veh'];
    $$total_hp_low_use_year_var = $row['total_hp'];
}

//====================================================================
=========
// vehicles sold / added

    $query = "SELECT COUNT(ein) as num_veh_sold_2006_2011
            FROM temp_2010
            WHERE !low_use
              AND (date_veh_sale > '2006-03-01')
              AND (date_veh_sale < '2011-03-01')";
```

```
    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
    $num_veh_sold_2006_2011 = $row['num_veh_sold_2006_2011'];

    $query = "SELECT COUNT(ein) as num_veh_added_2006_2009
            FROM temp_2010
            WHERE !low_use
              AND date_veh_purchase >= '2006-03-01'
              AND date_veh_purchase < '2009-03-02'";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
    $num_veh_added_2006_2009 = $row['num_veh_added_2006_2009'];

    $query = "SELECT COUNT(ein) as num_veh_added_2009_2010
            FROM temp_2010
            WHERE !low_use
              AND date_veh_purchase >= '2009-03-01'
              AND date_veh_purchase < '2010-03-02'";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
    $num_veh_added_2009_2010 = $row['num_veh_added_2009_2010'];

    $query = "SELECT COUNT(ein) as num_veh_added_2010_2011
            FROM temp_2010
            WHERE !low_use
              AND date_veh_purchase >= '2010-03-01'
              AND date_veh_purchase < '2011-03-02'";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
    $num_veh_added_2010_2011 = $row['num_veh_added_2010_2011'];

//============================================================================
========

    $retirement_2006_2010_credit = max($total_hp_fleet_average_year_2006
                              -$total_hp_fleet_average_year_2010, 0);

//============================================================================
===========
// Low-use vehicles

    $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                    COUNT(ein) as num_veh
```

```
                  FROM temp_2010
                  WHERE low_use
                    AND date_veh_sale IS NULL
                    AND date_eng_sale IS NULL";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
    $total_hp_low_use = $row['total_hp'];
    $num_veh_low_use = $row['num_veh'];

//==================================================================
===========
// Repowers

    $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                    COUNT(ein) as num_veh
            FROM temp_2010
            WHERE !low_use
              AND date_veh_sale IS NULL
              AND date_eng_sale IS NOT NULL";

    $result = @mysqli_query ($dbc, $query);

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
    $total_hp_repower = $row['total_hp'];
    $num_veh_repower = $row['num_veh'];

//==================================================================
===========
// Retrofits

    $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                    COUNT(ein) as num_veh
                  FROM temp_2010
                  WHERE vdecs_family_name IS NOT NULL
                    AND date_veh_sale IS NULL
                    AND date_eng_sale IS NULL";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
    $total_hp_retrofit = $row['total_hp'];
    $num_veh_retrofit = $row['num_veh'];

//==================================================================
===========
```

```
// Alt-fuel vehicles

    $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                        COUNT(ein) as num_veh
                    FROM temp_2010
                    WHERE alt_fuel
                      AND date_veh_sale IS NULL
                      AND date_eng_sale IS NULL";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
    $total_hp_alt_fuel = $row['total_hp'];
    $num_veh_alt_fuel = $row['num_veh'];

//=================================================================
===========
// SOON Moyer

    $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                        COUNT(ein) as num_veh
                    FROM temp_2010
                    WHERE program
                      AND date_veh_sale IS NULL
                      AND date_eng_sale IS NULL";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
    $total_hp_program = $row['total_hp'];
    $num_veh_program = $row['num_veh'];

//=================================================================
=========
// Tier 0s

    $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                        COUNT(ein) as num_veh
              FROM temp_2010
              LEFT JOIN zengine_index
                  USING (eng_model_year)
              WHERE date_veh_sale IS NULL
                AND eng_hp >= eng_index_min_hp
                AND eng_hp <= eng_index_max_hp
                AND eng_index_tier = 'T0'";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
```

```
   if (mysqli_num_rows($result)>0)
   {
      $total_hp_t0 = $row['total_hp'];
      $num_veh_t0 = $row['num_veh'];
   }

//==================================================================
=========
// Tier 1s

   $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                     COUNT(ein) as num_veh
          FROM temp_2010
          LEFT JOIN zengine_index
             USING (eng_model_year)
          WHERE date_veh_sale IS NULL
            AND eng_hp >= eng_index_min_hp
            AND eng_hp <= eng_index_max_hp
            AND eng_index_tier = 'T1U'";

   $result = @mysqli_query ($dbc, $query);
   $row = mysqli_fetch_assoc($result);

   if (mysqli_num_rows($result)>0)
   {
      $total_hp_t1U = $row['total_hp'];
      $num_veh_t1U = $row['num_veh'];
   }

//==================================================================
=========
// Tier 1Zs

   $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                     COUNT(ein) as num_veh
          FROM temp_2010
          LEFT JOIN zengine_index
             USING (eng_model_year)
          WHERE date_veh_sale IS NULL
            AND eng_hp >= eng_index_min_hp
            AND eng_hp <= eng_index_max_hp
            AND eng_index_tier = 'T1Z'";

   $result = @mysqli_query ($dbc, $query);
   $row = mysqli_fetch_assoc($result);
```

```
    if (mysqli_num_rows($result)>0)
    {
        $total_hp_t1Z = $row['total_hp'];
        $num_veh_t1Z = $row['num_veh'];
    }

//===================================================================
=========
// Tier 2s

    $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                        COUNT(ein) as num_veh
              FROM temp_2010
              LEFT JOIN zengine_index
                 USING (eng_model_year)
              WHERE date_veh_sale IS NULL
                AND eng_hp >= eng_index_min_hp
                AND eng_hp <= eng_index_max_hp
                AND eng_index_tier = 'T2'";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);

    if (mysqli_num_rows($result)>0)
    {
        $total_hp_t2 = $row['total_hp'];
        $num_veh_t2 = $row['num_veh'];
    }

//===================================================================
=========
// Tier 3s

    $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                        COUNT(ein) as num_veh
              FROM temp_2010
              LEFT JOIN zengine_index
                 USING (eng_model_year)
              WHERE date_veh_sale IS NULL
                AND eng_hp >= eng_index_min_hp
                AND eng_hp <= eng_index_max_hp
                AND eng_index_tier = 'T3'";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
```

```
    if (mysqli_num_rows($result)>0)
    {
        $total_hp_t3 = $row['total_hp'];
        $num_veh_t3 = $row['num_veh'];
    }

//===================================================================
=========
// Tier 4Is

    $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp,
                        COUNT(ein) as num_veh
            FROM temp_2010
            LEFT JOIN zengine_index
                USING (eng_model_year)
            WHERE date_veh_sale IS NULL
              AND eng_hp >= eng_index_min_hp
              AND eng_hp <= eng_index_max_hp
              AND eng_index_tier = 'T4I'";

    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);

    if (mysqli_num_rows($result)>0)
    {
        $total_hp_t4I = $row['total_hp'];
        $num_veh_t4I = $row['num_veh'];
    }

//===================================================================
=========
// average fleet age of vehicles (hp) in fleet average

$query = "SELECT
            SUM(IF(veh_model_year<2011,(2010-veh_model_year),0))
                        /COUNT(ein) as fleet_veh_age_by_count,
            SUM(IF(veh_model_year<2011,(2010-veh_model_year),0)*eng_hp)
                        /SUM(eng_hp) as fleet_veh_age_by_hp,
            SUM(IF(eng_model_year<2011,(2010-eng_model_year),0))
                        /COUNT(ein) as fleet_eng_age_by_count,
            SUM(IF(eng_model_year<2011,(2010-eng_model_year),0)*eng_hp)
                        /SUM(eng_hp) as fleet_eng_age_by_hp
        FROM temp_2010
        WHERE !low_use
          AND date_veh_sale IS NULL
```

```
          AND date_eng_sale IS NULL";

$result = @mysqli_query ($dbc, $query);
$row = mysqli_fetch_assoc($result);
$fleet_veh_age_by_count = $row['fleet_veh_age_by_count'];
$fleet_veh_age_by_hp    = $row['fleet_veh_age_by_hp'];
$fleet_eng_age_by_count = $row['fleet_eng_age_by_count'];
$fleet_eng_age_by_hp    = $row['fleet_eng_age_by_hp'];

//==================================================================
========
// Start of the reduced activity calculations

   // July 2007 hp
   $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as
july_2007_hp
            FROM temp_2010
            WHERE !low_use
              AND date_veh_purchase < '2007-07-01'
              AND ((date_veh_sale IS NULL)
                 OR (date_veh_sale > '2007-07-01'))
              AND date_eng_installed < '2007-07-01'
              AND ((date_eng_sale IS NULL)
                 OR (date_eng_sale > '2007-07-01'))";
   $result = @mysqli_query ($dbc, $query);
   $row = mysqli_fetch_assoc($result);
   $july_2007_hp = $row['july_2007_hp'];

// Reduced activity method
$query = "SELECT *
         FROM rentity_reduced_activity
         WHERE doors_id = '$doors_id'";

$result = @mysqli_query ($dbc, $query);
if (mysqli_num_rows($result)>0)
{
   $row = mysqli_fetch_assoc($result);

   $ren_red_act_method = $row['ren_red_act_method'];

   if($ren_red_act_method == '')
   {
      $ren_red_act_method = 'NA';
   }

   $ren_red_act_2007   = $row['ren_red_act_2007'];
```

```
    $ren_red_act_2010   = $row['ren_red_act_2010'];

    if($ren_red_act_2007 > 0)
    {
       $ren_red_act_pct = ($ren_red_act_2007 - $ren_red_act_2010)*100
                       / $ren_red_act_2007;
    }

    if ($ren_red_act_method == 'revenue'
       or $ren_red_act_method == 'employment'
       or $ren_red_act_method == 'production'
       or $ren_red_act_method == 'other')
    {
       $ren_red_act_pct =  min($ren_red_act_pct, 20);
    }

    $ren_red_act_pct =  number_format(max($ren_red_act_pct, 0), 1,".",",");

    // March 2010 hp
    $query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as
march_2010_hp
           FROM temp_2010
           WHERE !low_use
             AND date_veh_purchase < '2010-03-01'
             AND (date_veh_sale IS NULL)
             AND date_eng_installed < '2010-03-01'
             AND (date_eng_sale IS NULL)";
    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);
    $march_2010_hp = $row['march_2010_hp'];

    // average_2007_hp_hrs
    $query = "SELECT IF(veh_red_act_hours_2007 IS NOT NULL,
                   (SUM(veh_red_act_hours_2007)/COUNT(veh_red_act_hours_2007)
                  * $retirement_2006_2010_credit), 0) as average_2007_hp_hrs
         FROM temp_2010
         LEFT JOIN vehicle_reduced_activity
           USING (ein)
         WHERE !low_use
           AND (date_veh_sale < '2010-03-31')
           AND (date_eng_sale IS NULL
             OR date_eng_sale > '2010-03-31')";

    $result = @mysqli_query ($dbc, $query);
    if (mysqli_num_rows($result)>0)
    {
```

```
    $row = mysqli_fetch_assoc($result);
    $average_2007_hp_hrs = $row['average_2007_hp_hrs'];
}

// activity_2007
$query = "SELECT IF(veh_red_act_hours_2007 IS NOT NULL,
                    (SUM(eng_hp*veh_red_act_hours_2007)
                     - $average_2007_hp_hrs), 0) as activity_2007
      FROM temp_2010
      LEFT JOIN vehicle_reduced_activity
        USING (ein)
      WHERE !low_use
        AND date_veh_purchase < '2008-01-01'
        AND (date_veh_sale IS NULL
          OR date_veh_sale > '2007-01-01')
        AND date_eng_installed < '2008-01-01'
        AND (date_eng_sale IS NULL
          OR date_eng_sale > '2007-01-01')";

$result = @mysqli_query ($dbc, $query);
if (mysqli_num_rows($result)>0)
{
    $row = mysqli_fetch_assoc($result);
    $activity_2007 = $row['activity_2007'];
}

// average 2007 hp
$query = "SELECT ROUND(SUM(eng_hp*
            if( (date_veh_sale > '2007-01-01')
                    and (date_veh_sale<'2007-12-31'),
                 (DATE_FORMAT(date_veh_sale, '%j')
                  - DATE_FORMAT('2007-01-01', '%j'))
                         /365,1)), 0)
                as average_2007_hp
          FROM temp_2010
          WHERE !low_use
            AND date_veh_purchase < '2007-01-01'
            AND (date_veh_sale IS NULL
              OR date_veh_sale > '2007-01-01')
            AND date_eng_installed < '2007-01-01'
            AND (date_eng_sale IS NULL
              OR date_eng_sale > '2007-01-01')";

$result = @mysqli_query ($dbc, $query);
$row = mysqli_fetch_assoc($result);
$average_2007_hp = $row['average_2007_hp'];
```

```
   // activity_2010
   $query = "SELECT IF(veh_red_act_hours_2010 IS NOT NULL,
                  SUM(eng_hp*veh_red_act_hours_2010), 0) as activity_2010
      FROM temp_2010
      LEFT JOIN vehicle_reduced_activity
        USING (ein)
      WHERE !low_use
        AND date_veh_purchase < '2010-04-01'
        AND (date_veh_sale IS NULL
          OR date_veh_sale >= '2009-03-01')
        AND date_eng_installed < '2010-04-01'
        AND (date_eng_sale IS NULL
          OR date_eng_sale >= '2009-03-01')";

   $result = @mysqli_query ($dbc, $query);
   if (mysqli_num_rows($result)>0)
   {
      $row = mysqli_fetch_assoc($result);
      $activity_2010 = $row['activity_2010'];
   }

   $reduced_activity_credit = 0;
//======================================================================
========

   if ($ren_red_act_method == 'hours')
   {
      // check to see if the owner has entered hours of operation for
      // at least 95% of their 2010 fleet hp
      $query = "SELECT SUM(eng_hp) as hp_pool
            FROM rentity_vehicle
            LEFT JOIN vehicle_reduced_activity
              USING (doors_id, ein)
            LEFT JOIN vehicle
               USING (ein)
            LEFT JOIN zvehicle_man_mod
               USING (veh_man_mod_id)
            LEFT JOIN vehicle_engine
               USING (ein)
            LEFT JOIN engine_hp
               USING (engine_id)
            WHERE rentity_vehicle.doors_id = $doors_id
              AND (ISNULL(date_veh_sale)
                 OR date_veh_sale >= '2010-03-01')";
```

```
$result = @mysqli_query ($dbc, $query);
if (mysqli_num_rows($result)>0)
{
   $row = mysqli_fetch_assoc($result);
   $hp_pool = $row['hp_pool'];
}

$query = "SELECT SUM(eng_hp) as hp_entered
        FROM rentity_vehicle
        IINER JOIN vehicle_reduced_activity
          USING (doors_id, ein)
        LEFT JOIN vehicle
           USING (ein)
        LEFT JOIN zvehicle_man_mod
           USING (veh_man_mod_id)
        LEFT JOIN vehicle_engine
           USING (ein)
        LEFT JOIN engine_hp
           USING (engine_id)
        WHERE doors_id = $doors_id
          AND NOT ISNULL(veh_red_act_hours_2010)
          AND (veh_red_act_hours_2010 != '')
          AND (ISNULL(date_veh_sale)
               OR date_veh_sale >= '2010-03-01')";

$result = @mysqli_query ($dbc, $query);
if (mysqli_num_rows($result)>0)
{
   $row = mysqli_fetch_assoc($result);
   $hp_entered = $row['hp_entered'];
}

$not_enough_hp = 0;

if ($hp_pool > 0)
{
   if(($hp_entered / $hp_pool) < 0.95)
   {
      $ren_red_act_pct = 0;
      $reduced_activity_credit = 0;
   }
   else
   {
      if ($activity_2007 > 0)
      {  // see line 153
         $ren_red_act_pct = ($activity_2007 - $activity_2010)*100
```

```
                                    / $activity_2007;

                $ren_red_act_pct =  number_format(max($ren_red_act_pct, 0), 1,".",",");

                $reduced_activity_credit = $ren_red_act_pct/100*($july_2007_hp
                                   - min($retirement_2006_2010_credit,
                                     $average_2007_hp-$march_2010_hp));

            }
          }
       }
    }
    else
    {
       $reduced_activity_credit = $ren_red_act_pct/100*$july_2007_hp
                             - $retirement_2006_2010_credit;
    }

    $reduced_activity_credit = max($reduced_activity_credit, 0);
}

//==================================================================
=========
// Early Replacement Credit - Turnover (Horsepower)

// Tier 0 vehicles retired before 2009-03-01
$query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp
         FROM temp_2010
         LEFT JOIN zengine_index
            USING (eng_model_year)
         WHERE !low_use
           AND date_veh_purchase < '2006-03-01'
           AND (date_veh_sale < '2009-03-01')
           AND date_eng_installed < '2006-03-01'
           AND (date_eng_sale IS NULL
             OR date_eng_sale > '2009-03-01')
           AND eng_hp >= eng_index_min_hp
           AND eng_hp <= eng_index_max_hp
           AND eng_index_tier = 'T0'";

$result = @mysqli_query ($dbc, $query);

if (mysqli_num_rows($result)>0)
{
    $row = mysqli_fetch_assoc($result);
    $early_replacement_credit = $row['total_hp'];
```

```
}

for ($year_query=2007 ; $year_query<2010; $year_query++ )
{
   $total_hp_fleet_average_year_var = 'total_hp_fleet_average_year_' . $year_query;
   $early_replacement_credit -= $$total_hp_fleet_average_year_var * 0.08;
}

$hp_reduction_06_09 = max($total_hp_fleet_average_year_2006
                     - $total_hp_fleet_average_year_2009, 0);
$hp_reduction_06_10 = max($total_hp_fleet_average_year_2006
                     - $total_hp_fleet_average_year_2010, 0);

$early_replacement_credit -= $hp_reduction_06_10;

$early_replacement_credit = max($early_replacement_credit,0);


//==================================================================
=========
// vehicles retired in 2009-2010 turnover

// find 09-10 hp reduction
$query = "SELECT  IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp
        FROM temp_2010
        WHERE !low_use
          AND date_veh_purchase < '2009-03-01'
          AND (date_veh_sale >= '2009-03-01')
          AND (date_veh_sale < '2010-03-01')
          AND date_eng_installed < '2009-03-01'
          AND (date_eng_sale IS NULL
            OR date_eng_sale > '2010-03-01')";

$result = @mysqli_query ($dbc, $query);

if (mysqli_num_rows($result)>0)
{
   $row = mysqli_fetch_assoc($result);
   $hp_reduction_09_10 = $row['total_hp'];
}

// find lowest tier in 2009 fleet
$query = "SELECT DISTINCT eng_index_tier
        FROM temp_2010
        LEFT JOIN zengine_index
            USING (eng_model_year)
```

```
        WHERE !low_use
          AND date_veh_purchase < '2009-03-01'
          AND (date_veh_sale IS NULL
            OR date_veh_sale < '2010-03-01')
          AND date_eng_installed < '2009-03-01'
          AND (date_eng_sale IS NULL
            OR date_eng_sale > '2010-03-01')
          AND eng_hp >= eng_index_min_hp
          AND eng_hp <= eng_index_max_hp
        ORDER BY eng_index_tier";

$result = @mysqli_query ($dbc, $query);
$row = mysqli_fetch_assoc($result);
$lowest_tier_2009 = $row['eng_index_tier'];

// lowest tier hp reduction
$query = "SELECT  IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp
        FROM temp_2010
        LEFT JOIN zengine_index
            USING (eng_model_year)
        WHERE !low_use
          AND date_veh_purchase < '2009-03-01'
          AND (date_veh_sale >= '2009-03-01')
          AND (date_veh_sale < '2010-03-01')
          AND date_eng_installed < '2009-03-01'
          AND (date_eng_sale IS NULL
            OR date_eng_sale > '2010-03-01')
          AND eng_hp >= eng_index_min_hp
          AND eng_hp <= eng_index_max_hp
          AND (eng_index_tier = 'T0'
            OR eng_index_tier = 'T1U'
            OR eng_index_tier <= '$lowest_tier_2009')";

$result = @mysqli_query ($dbc, $query);

if (mysqli_num_rows($result)>0)
{
    $row = mysqli_fetch_assoc($result);
    $lowest_tier_hp_reduction_09_10 = $row['total_hp'];
}

$turnover09_10 = $lowest_tier_hp_reduction_09_10
                - min($hp_reduction_09_10, $retirement_2006_2010_credit);

$turnover09_10 = max($turnover09_10,0);
```

```
//=====================================================================
=========
// Tier 0 retirement VDECS credit

// tier 0 hp reduction 09-10
$query = "SELECT  IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp
         FROM temp_2010
         LEFT JOIN zengine_index
            USING (eng_model_year)
         WHERE !low_use
           AND date_veh_purchase < '2009-03-01'
           AND (date_veh_sale >= '2009-03-01')
           AND (date_veh_sale < '2010-03-01')
           AND date_eng_installed < '2009-03-01'
           AND (date_eng_sale IS NULL
             OR date_eng_sale > '2010-03-01')
           AND eng_hp >= eng_index_min_hp
           AND eng_hp <= eng_index_max_hp
           AND eng_index_tier = 'T0'";

$result = @mysqli_query ($dbc, $query);

if (mysqli_num_rows($result)>0)
{
    $row = mysqli_fetch_assoc($result);
    $tier0_hp_reduction_09_10 = $row['total_hp'];
}

$tier0_retirement_vdecs_credit = min($tier0_hp_reduction_09_10,
                                   $hp_reduction_09_10)
                              - min($retirement_2006_2010_credit,
                                    $hp_reduction_09_10);

$tier0_retirement_vdecs_credit = max($tier0_retirement_vdecs_credit, 0);


//=====================================================================
=========
// Repowers

$query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp
         FROM temp_2010
         WHERE !low_use
           AND date_veh_purchase < '2010-03-01'
           AND date_veh_sale IS NULL
           AND date_eng_sale IS NOT NULL";
```

```
$result = @mysqli_query ($dbc, $query);

if (mysqli_num_rows($result)>0)
{
   $row = mysqli_fetch_assoc($result);
   $repower_hp = $row['total_hp'];
}


//==================================================================
=========
// Double Retrofit for NOx retrofits

$query = "SELECT IF(SUM(eng_hp) IS NOT NULL,
                  SUM(eng_hp*(1-vdecs_nox)*100)/60, 0) as total_hp
        FROM temp_2010
        WHERE !low_use
          AND date_vdecs_purchase IS NOT NULL
          AND date_vdecs_purchase < '2010-03-01'
          AND date_veh_sale IS NULL
          AND date_eng_sale IS NULL";

$result = @mysqli_query ($dbc, $query);

if (mysqli_num_rows($result)>0)
{
   $row = mysqli_fetch_assoc($result);
   $double_nox_retrofit_credit = $row['total_hp']*2;
}

//==================================================================
=========
// Permanent low-use NOx credit

$query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp
        FROM temp_2010
        INNER JOIN vehicle_low_use
          USING (ein)
        WHERE low_use
          AND (veh_low_use_type = 'designated'
              OR  veh_low_use_type = 'designated_outside')
          AND date_veh_sale IS NULL
          AND date_eng_sale IS NULL";

$result = @mysqli_query ($dbc, $query);
```

```
if (mysqli_num_rows($result)>0)
{
   $row = mysqli_fetch_assoc($result);
   $permanent_low_use_nox_credit = $row['total_hp'];
}

//==================================================================
=========
// Double Retrofit

$query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp
        FROM temp_2010
        WHERE !low_use
          AND date_vdecs_purchase IS NOT NULL
          AND date_vdecs_purchase < '2010-01-01'
          AND date_veh_sale IS NULL
          AND date_eng_sale IS NULL";

$result = @mysqli_query ($dbc, $query);

if (mysqli_num_rows($result)>0)
{
   $row = mysqli_fetch_assoc($result);
   $double_retrofit_credit = $row['total_hp']*2;
}

//==================================================================
=========
// Single Retrofit

$query = "SELECT IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) as total_hp
        FROM temp_2010
        WHERE !low_use
          AND date_vdecs_purchase IS NOT NULL
          AND date_vdecs_purchase > '2010-03-01'
          AND date_veh_sale IS NULL
          AND date_eng_sale IS NULL";

$result = @mysqli_query ($dbc, $query);

if (mysqli_num_rows($result)>0)
{
   $row = mysqli_fetch_assoc($result);
   $single_retrofit_credit = $row['total_hp'];
}
```

```
//==================================================================
=========

$nox_credit =   $retirement_2006_2010_credit
            + $reduced_activity_credit
            + $early_replacement_credit
            + $repower_hp
            + $turnover09_10
            + $double_nox_retrofit_credit
            + $permanent_low_use_nox_credit;

$pm_credit =  $retirement_2006_2010_credit
          + $reduced_activity_credit
          + $tier0_retirement_vdecs_credit
          + $double_retrofit_credit
          + $single_retrofit_credit;

//==================================================================
=========
// bact

$nox_bact_hp = $total_hp_fleet_average_year_2009 * 0.08;
$pm_bact_hp  = $total_hp_fleet_average_year_2009 * 0.2;

//==================================================================
===========
// get 2010 fleet averages, see write_fleet_average() in doors_function


    $query = "SELECT
              IF(SUM(eng_hp) IS NOT NULL,
                 SUM(eng_hp*eng_index_nox*vdecs_nox)/SUM(eng_hp),0)
                      AS nox_fleet_average,
              IF(SUM(eng_hp) IS NOT NULL,
                 SUM(eng_hp*eng_index_pm*vdecs_pm)/SUM(eng_hp),0)
                      AS pm_fleet_average

             FROM temp_2010
             INNER JOIN zengine_index
                USING (eng_model_year)
             WHERE !low_use
              AND date_veh_sale IS NULL
              AND date_eng_sale IS NULL
              AND (zengine_index.eng_index_min_hp <= eng_hp)
              AND (zengine_index.eng_index_max_hp >= eng_hp)";
```

```
      $result = @mysqli_query ($dbc, $query);

   if(mysqli_num_rows($result)>0)
   {
      $row = mysqli_fetch_assoc($result);

      $nox_fleet_average_2010  = number_format($row['nox_fleet_average'], 1);
      $pm_fleet_average_2010   = number_format($row['pm_fleet_average'], 2);
   }

//=====================================================================
===========
// determine fleet targets

/*
   $query = "SELECT
           ROUND(Sum(eng_hp*eng_target_nox)
                          /IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) , 1)
                   AS nox_fleet_target,
           ROUND(Sum(eng_hp*eng_target_pm)
                          /IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) , 2)
                   AS pm_fleet_target
           FROM zengine_target, temp_2010
           INNER JOIN zengine_index
              USING (eng_model_year)
           WHERE !low_use
             AND date_veh_sale IS NULL
             AND date_eng_sale IS NULL
                AND (zengine_target.eng_target_min_hp <= eng_hp)
                AND (zengine_target.eng_target_max_hp >= eng_hp)
                AND (eng_target_fleet_size = '$ren_size')
             AND target_year = '2010'";
   $result = @mysqli_query ($dbc, $query);
   $row = mysqli_fetch_assoc($result);

   $nox_fleet_target_2010   = $row['nox_fleet_target'];
   $pm_fleet_target_2010    = $row['pm_fleet_target'];
*/

//=====================================================================
==========
// determine fleet targets


$begin_year = 2010;
$target_year = 2020;
```

```
if ($ren_size !== 'S')
{
   for ($i=$begin_year; $i<=$target_year; $i++)
   {
      $query = "SELECT
               IF(SUM(eng_hp) IS NOT NULL,
                     SUM(eng_hp*eng_target_nox)/SUM(eng_hp),0)
                      AS nox_fleet_target
              FROM zengine_target, temp_2010
              INNER JOIN zengine_index
                 USING (eng_model_year)
              WHERE !low_use
                AND date_veh_sale IS NULL
                AND date_eng_sale IS NULL
                   AND (zengine_target.eng_target_min_hp <= eng_hp)
                   AND (zengine_target.eng_target_max_hp >= eng_hp)
                   AND (eng_target_fleet_size = '$ren_size')
             AND calendar_year = '$i'";

      $result = @mysqli_query ($dbc, $query);
      $row = mysqli_fetch_assoc($result);

      $nox_fleet_target_year = 'nox_fleet_target_' . $i;

      $$nox_fleet_target_year =  number_format($row['nox_fleet_target'], 1);
   }
}

for ($i=$begin_year; $i<=$target_year; $i++)
{
   $query = "SELECT
               IF(SUM(eng_hp) IS NOT NULL,
                     SUM(eng_hp*eng_target_pm)/SUM(eng_hp),0)
                                     AS pm_fleet_target
           FROM zengine_target, temp_2010
           INNER JOIN zengine_index
              USING (eng_model_year)
           WHERE !low_use
             AND date_veh_sale IS NULL
             AND date_eng_sale IS NULL
                AND (zengine_target.eng_target_min_hp <= eng_hp)
                AND (zengine_target.eng_target_max_hp >= eng_hp)
                AND (eng_target_fleet_size = '$ren_size')
             AND calendar_year = '$i'";
```

```
    $result = @mysqli_query ($dbc, $query);
    $row = mysqli_fetch_assoc($result);

    $pm_fleet_target_year = 'pm_fleet_target_' . $i;

    $$pm_fleet_target_year =  number_format($row['pm_fleet_target'], 2);
}

//=====================================================================
=========

    if ($ren_size=='S'
       or $ren_size=='M')
    {
       $meets_nox_fleet_average  = 'NA';
       $meets_nox_bact          = 'NA';
       $meets_nox              = 'Met';

       $meets_pm_fleet_average   = 'NA';
       $meets_pm_bact           = 'NA';
       $meets_pm               = 'Met';
    }
    else
    {


//=====================================================================
===========
    // set compliance flags

       $meets_nox_fleet_average  = 'Not Met';
       $meets_nox_bact          = 'Not Met';
       $meets_nox              = 'Not Met';

       $meets_pm_fleet_average   = 'Not Met';
       $meets_pm_bact           = 'Not Met';
       $meets_pm               = 'Not Met';

       if($nox_bact_hp <= $nox_credit)
       {
          $meets_nox_bact = 'Met';
          $meets_nox      = 'Met';
       }

       if($nox_fleet_average_2010 <= $nox_fleet_target_2010)
```

```
        {
          $meets_nox_fleet_average  = 'Met';
          $meets_nox_bact           = 'NA';
          $meets_nox                = 'Met';
        }

        if($pm_bact_hp <= $pm_credit)
        {
          $meets_pm_bact  = 'Met';
          $meets_pm       = 'Met';
        }

        if($pm_fleet_average_2010 <= $pm_fleet_target_2010)
        {
          $meets_pm_fleet_average = 'Met';
          $meets_pm_bact          = 'NA';
          $meets_pm               = 'Met';
        }
    }
    }
?>
```

```
A<?php  #get_fleet_average

$query3 = "SELECT
        ROUND(Sum(eng_hp*eng_index_nox*vdecs_nox)
                        /IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) , 1)
                AS fleet_average_nox
        FROM osm_vehicle
        WHERE calendar_year = '$calendar_year'";

$result3 = @mysqli_query ($dbc4, $query3);
$row3 = mysqli_fetch_assoc($result3);

$fleet_average_nox =  number_format($row3['fleet_average_nox'], 1);


$query3 = "SELECT
        ROUND(Sum(eng_hp*eng_index_pm*vdecs_pm)
                         /IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) , 2)
                 AS fleet_average_pm
        FROM osm_vehicle
        WHERE calendar_year = '$calendar_year'";
```

```
$result3 = @mysqli_query ($dbc4, $query3);
$row3 = mysqli_fetch_assoc($result3);

$fleet_average_pm =  number_format($row3['fleet_average_pm'], 2);

?>
```

```
<?php  #get_fleet_info


$query0 = "SELECT SUM(eng_hp)    AS total_hp,
                  COUNT(*)       AS veh_count,
                  SUM(eng_hp*(calendar_year-
                         IF(eng_model_year=2011 AND calendar_year=2010,
                                    2010,eng_model_year)))
                              / SUM(eng_hp)
                         AS fleet_age
           FROM osm_vehicle
           WHERE doors_id = $doors_id
             AND calendar_year = $calendar_year
             AND low_use = 0";

$result0 = @mysqli_query ($dbc4, $query0);
$row0 = mysqli_fetch_assoc($result0);
$veh_count = $row0['veh_count'];

if ($veh_count == 0)
{
   $total_hp          = 0;
   $fleet_age          = 0;
}
else
{
   $total_hp          = $row0['total_hp'];
   $fleet_age          = $row0['fleet_age'];

   if($calendar_year == '2010')
   {
      $fleet_age_2010     = $fleet_age;
      $fleet_age_adjusted = $fleet_age;

      $query_fleet = "SELECT veh_type
                      FROM osm_vehicle
                      WHERE doors_id = '$doors_id'
                        AND calendar_year = '2010'
```

```
                    AND '2010' - calendar_year_retrofit >= 6
                    AND low_use = 0
                   ORDER BY veh_relative_age DESC, eng_index_tier, ein";

      if(!$result = @mysqli_query ($dbc4, $query_fleet)) $message .= '_59';

      $row = mysqli_fetch_assoc($result);
      $veh_type = $row['veh_type'];

      $query4 = "SELECT veh_average_age as replacement_vehicle_age
               FROM zvehicle_average_age
               WHERE veh_type = '$veh_type'";
        $result4 = @mysqli_query ($dbc4, $query4);

      $row4 = mysqli_fetch_assoc($result4);
      $replacement_vehicle_age= $row4['replacement_vehicle_age'];

      if(fmod($doors_id, 2)>0)
      {
         $fleet_age_delta = 2*(($fleet_age_2010
                           - round($replacement_vehicle_age))
                             /$veh_count);
      }
   }
}

?>
```

```
<?php  #get_fleet_target

if ($run==0)
{
   $fleet_target_nox=0;
   $fleet_target_pm=0;
}
else
{
   $query = "SELECT
          ROUND(Sum(eng_hp*eng_target_nox)
                        /IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) , 1)
                  AS fleet_target_nox
          FROM osm_vehicle
          WHERE calendar_year = '$calendar_year'";
```

```
   $result = @mysqli_query ($dbc4, $query);
   $row = mysqli_fetch_assoc($result);

   $fleet_target_nox =  number_format($row['fleet_target_nox'], 1);


   $query = "SELECT
           ROUND(Sum(eng_hp*eng_target_pm)
                                /IF(SUM(eng_hp) IS NOT NULL, SUM(eng_hp), 0) , 2)
                       AS fleet_target_pm
           FROM osm_vehicle
           WHERE calendar_year = '$calendar_year'";

   $result = @mysqli_query ($dbc4, $query);
   $row = mysqli_fetch_assoc($result);

   $fleet_target_pm =  number_format($row['fleet_target_pm'], 2);
}

?>
```

```
<?php  # htm_initialize  ?>


<!--
=====================================================================
======-->

      <TD colspan="2"><FIELDSET><legend><FONT
COLOR="RED"><b>INITIALIZE</b></FONT></legend>

     <TABLE BORDER="0" cellpadding="2" CELLSPACING="1"  width="100%"
bgcolor="#D0D0D0">

        <TR><TD>
            Run base
           </TD><TD>
           <INPUT TYPE="CHECKBOX" NAME="run_base" CHECKED>
        </TD></TR>

        <TR><TD>
            Run scenario
           </TD><TD>
           <INPUT TYPE="CHECKBOX" NAME="run_scenario" CHECKED>
```

```
        </TD></TR>

        <TR><TD>
           End year
           </TD><TD><SELECT NAME="end_year">
<?php          for ($i=2011; $i<= 2030; $i++)
           {
              if (isset($end_year) and ($end_year == $i))
              {
                 echo '<OPTION  SELECTED="SELECTED">' . $i . '</OPTION>';
              }
              else
              {
                 echo '<OPTION>' . $i . '</OPTION>';
              }
           }
?>
          </SELECT>
        </TD></TR>

        <TR><TD>
           Fleet target file name
           </TD><TD>
           <INPUT TYPE="TEXT" NAME="target_file_name"
           SIZE="26" MAXLENGTH="25"
           <?php if(isset($target_file_name)) echo ' VALUE="' . $target_file_name . '"' ?>
           >
        </TD></TR>

  <!--      <TR><TD>
            NOx targets only, retrofit credit
           </TD><TD>
           <INPUT TYPE="CHECKBOX" NAME="nox_targets_only"
           <?php if($nox_targets_only) echo ' CHECKED="CHECKED"' ?>
           >
        </TD></TR> -->

        <TR><TD>
           Vehicle cost file name
           </TD><TD>
           <INPUT TYPE="TEXT" NAME="vehicle_cost_file_name"
           SIZE="26" MAXLENGTH="25"
           <?php if(isset($vehicle_cost_file_name)) echo ' VALUE="' .
$vehicle_cost_file_name . '"' ?>
           >
        </TD></TR>
```

```
        <TR><TD>
             Do cost after run simulation
           </TD><TD>
           <INPUT TYPE="CHECKBOX" NAME="link_cost"
           <?php if($link_cost) echo ' CHECKED="CHECKED"' ?>
           >
        </TD></TR>

        <TR><TD>Economic Recovery
        </TD><TD>
        Long term <INPUT TYPE="RADIO" NAME="economic_recovery"
value="long_term"
        checked="checked"
        >
        </TD></TR>

 <!--        <TR><TD colspan="2">
                NA <INPUT TYPE="RADIO"
NAME="economic_recovery" value="na"
        <?php
           if ($economic_recovery == 'na') echo 'checked = "checked"';
        ?>
        > 
        Slow <INPUT TYPE="RADIO" NAME="economic_recovery" value="slow"
        <?php
           if ($economic_recovery == 'slow') echo 'checked = "checked"';
        ?>
        > 
        Average <INPUT TYPE="RADIO" NAME="economic_recovery" value="average"
        <?php
           if ($economic_recovery == 'average') echo 'checked = "checked"';
        ?>
        > 
        Fast <INPUT TYPE="RADIO" NAME="economic_recovery" value="fast"
        <?php
           if ($economic_recovery == 'fast') echo 'checked = "checked"';
        ?>
              ></TD>
        </TR> -->

        <TR><TD>
           Run name
           </TD><TD>
           <INPUT TYPE="TEXT" NAME="run_name"
           SIZE="26" MAXLENGTH="35"
```

```
<?php

 if(isset($run_name)) echo ' VALUE="' . $run_name . '"' ?>
>
</TD></TR>

</TABLE>


</FIELDSET></TD>
```

```
<?php #intialize_base

if($fleet_age_2010 < 4)
{
	$replacement_vehicle_age_factor = 0;
}
if($fleet_age_2010 >= 4 And $fleet_age_2010 < 8)
{
	$replacement_vehicle_age_factor = 0;
}
if($fleet_age_2010 >= 8 And $fleet_age_2010 < 12)
{
	$replacement_vehicle_age_factor = 0.17;
}
if($fleet_age_2010 >= 12 And $fleet_age_2010 < 16)
{
	$replacement_vehicle_age_factor = 0.33;
}
if($fleet_age_2010 >= 16 And $fleet_age_2010 < 20)
{
	$replacement_vehicle_age_factor = 0.5;
}
if($fleet_age_2010 >= 20)
{
	$replacement_vehicle_age_factor = 0.75;
}

?>
```

```
<?php  #initialize_fleet
```

```
$calendar_year = 2010;

$fleet_age_delta=0;
$economic_recovery_factor=0;

$turnover_forced = 0;
$retrofit_forced = 0;

$turnover_carryover=0;
$retrofit_carryover=0;

$expiring_nox_credit = 0;
$expiring_pm_credit = 0;


?>
```

```
<?php  #initialize_run

$message = '';

$end_year              = 2011;
$target_file_name      = 'zengine_target';
$vehicle_cost_file_name = 'zvehicle_cost';
$run_name              = 'Run';
$economic_recovery     = 'long_term';

$turnover_only = 0;

$pm_credit_for_turnover_factor = 0;

$end_year_double_retrofit = 2010;

$new_vehicle_turnover_exemption_age = 10;

$discount_rate = 5.0;
$discount_year = 2010;
$cost_transport = 10;

$cost_retrofit_50 = 16750;
$cost_retrofit_125 = 17588;
$cost_retrofit_175 = 19733;
$cost_retrofit_300 = 24796;
$cost_retrofit_400 = 28763;
```

```
$cost_retrofit_600 = 52333;
$cost_retrofit_600_plus = 52333;

$cost_tier4_premium_50 = 8000;
$cost_tier4_premium_175 = 12000;
$cost_tier4_premium_400 = 18000;
$cost_tier4_premium_400_plus =30000;

$free_2012_for_2010=0;


if (isset($_POST['run_sim1']))
{
   $end_year              = 2030;
   $target_file_name       = 'zet_csx5';
}


?>
```

```
<?php  #initialize_scenario

if($fleet_age_2010 < 4)
{
      $replacement_vehicle_age_factor = 0;
}
if($fleet_age_2010 >= 4 And $fleet_age_2010 < 8)
{
      $replacement_vehicle_age_factor = 0;
}
if($fleet_age_2010 >= 8 And $fleet_age_2010 < 12)
{
      $replacement_vehicle_age_factor = 0.08;
}
if($fleet_age_2010 >= 12 And $fleet_age_2010 < 16)
{
      $replacement_vehicle_age_factor = 0.17;
}
if($fleet_age_2010 >= 16 And $fleet_age_2010 < 20)
{
      $replacement_vehicle_age_factor = 0.25;
}
if($fleet_age_2010 >= 20)
{
      $replacement_vehicle_age_factor = 0.33;
```

```
}

?>
```

```
<?php  #initialize_year

$turnover_carryover_begin = $turnover_carryover;
$turnover_needed=0;
$turnover_actual_base=0;
$turnover_actual=0;
$turnover_pct=0;
$turnover_pool_empty = 'No';


$turnover_tier=0;
$turnover_tier4=0;
$expiring_nox_credit_begin = $expiring_nox_credit;

$retrofit_carryover_begin = $retrofit_carryover;
$retrofit_needed=0;
$retrofit_actual=0;
$retrofit_pct=0;
$retrofit_pool_empty = 'No';

$expiring_pm_credit_begin = $expiring_pm_credit;

if($calendar_year==2010) $turnover_rate_forced = 0;
if($calendar_year==2011) $turnover_rate_forced = .048;
if($calendar_year==2012) $turnover_rate_forced = .048;
if($calendar_year==2013) $turnover_rate_forced = .144;
if($calendar_year==2014) $turnover_rate_forced = .08;
if($calendar_year==2015) $turnover_rate_forced = .08;

if($calendar_year>2015)
{
   $turnover_rate_forced = .1;
}

if($calendar_year==2010) $retrofit_rate_forced = 0;
if($calendar_year==2011) $retrofit_rate_forced = .12;
if($calendar_year==2012) $retrofit_rate_forced = .12;
if($calendar_year==2013) $retrofit_rate_forced = .36;

if($calendar_year>2013)
```

```
{
   $retrofit_rate_forced = .2;
}


if (isset($_POST['run_sim1']))
{

   if($calendar_year<=2013) $turnover_rate_forced = 0;
   if($calendar_year==2014) $turnover_rate_forced = .048;
   if($calendar_year>=2015 and $calendar_year<=2017) $turnover_rate_forced = .08;
   if($calendar_year>=2018) $turnover_rate_forced = .1;

   $retrofit_rate_forced = 0;
}

$annual_turnover_cost = 0;
$annual_repower_cost = 0;
$annual_full_turnover_cost = 0;
$annual_retrofit_cost = 0;

if($run==1 and $calendar_year == 2012)
{
   $expiring_nox_credit = 0;
   $expiring_nox_credit_begin = 0;
   $expiring_pm_credit = 0;
   $expiring_pm_credit_begin = 0;
}


?>
```

```
<?php #insert_engine_index_target

   $query4 = "UPDATE osm_vehicle
          INNER JOIN zengine_index
            USING (eng_model_year)
          SET osm_vehicle.eng_index_tier  = zengine_index.eng_index_tier,
              osm_vehicle.eng_index_pm    = zengine_index.eng_index_pm,
              osm_vehicle.eng_index_nox   = zengine_index.eng_index_nox
          WHERE (osm_vehicle.eng_hp >= zengine_index.eng_index_min_hp)
            AND (osm_vehicle.eng_hp <= zengine_index.eng_index_max_hp)
            AND ein = '$ein'";
   if (!$result4 = @mysqli_query ($dbc4, $query4)) $message .= '_166';
```

```
    $query4 = "UPDATE osm_vehicle
            INNER JOIN $target_file_name
              USING (calendar_year)
            SET osm_vehicle.eng_target_nox = $target_file_name.eng_target_nox,
                osm_vehicle.eng_target_pm  = $target_file_name.eng_target_pm
            WHERE   calendar_year      = '$calendar_year'
                AND osm_vehicle.eng_hp  >= $target_file_name.eng_target_min_hp
                AND osm_vehicle.eng_hp  <= $target_file_name.eng_target_max_hp
                AND $target_file_name.eng_target_fleet_size = '$ren_size'
              AND ein = '$ein'";
    if(!$result4 = @mysqli_query ($dbc4, $query4)) $message .=
'_insert_eng_index_target_27';


?>
```

```
<?php #insert_fleet_averages_targets

    $query4 = "UPDATE osm_vehicle
            INNER JOIN zengine_index
              USING (eng_model_year)
            SET osm_vehicle.eng_index_tier  = zengine_index.eng_index_tier,
                osm_vehicle.eng_index_pm    = zengine_index.eng_index_pm,
                osm_vehicle.eng_index_nox   = zengine_index.eng_index_nox
            WHERE (osm_vehicle.eng_hp >= zengine_index.eng_index_min_hp)
              AND (osm_vehicle.eng_hp <= zengine_index.eng_index_max_hp)";

    if (!$result4 = @mysqli_query ($dbc4, $query4)) $message .= '_166';


    $query4 = "UPDATE osm_vehicle
            INNER JOIN $target_file_name
              USING (calendar_year)
            SET osm_vehicle.eng_target_nox = $target_file_name.eng_target_nox,
                osm_vehicle.eng_target_pm  = $target_file_name.eng_target_pm
            WHERE   calendar_year      = '$calendar_year'
                AND osm_vehicle.eng_hp  >= $target_file_name.eng_target_min_hp
                AND osm_vehicle.eng_hp  <= $target_file_name.eng_target_max_hp
                AND $target_file_name.eng_target_fleet_size = '$ren_size'";

    if(!$result4 = @mysqli_query ($dbc4, $query4)) $message .= '_123';

?>
```

```
<?php #insert_vehicle_average_age

   $query4 = "UPDATE osm_vehicle
            INNER JOIN zvehicle_average_age
              USING ( veh_type )
            SET osm_vehicle.veh_average_age =
zvehicle_average_age.veh_average_age";
                  $result4 = @mysqli_query ($dbc4, $query4);

   $query4 = "UPDATE osm_vehicle
             SET veh_relative_age = (calendar_year - eng_model_year)/veh_average_age
             WHERE calendar_year     = '$calendar_year'";
                  $result4 = @mysqli_query ($dbc4, $query4);

?>
```

```
<?php # move_fleet_to_base_scenario

   if($run==0)
   {
      $query = 'REPLACE INTO osm_vehicle_base
                  SELECT *
                  FROM osm_vehicle';
      $result = mysqli_query($dbc4, $query);
   }
   else
   {
      $query = 'REPLACE INTO osm_vehicle_scenario
                  SELECT *
                  FROM osm_vehicle';
      $result = mysqli_query($dbc4, $query);
   }

?>
```

```
<?php  #run_sim

$query0 = "SELECT doors_id
         FROM rentity";
$result0 = @mysqli_query($dbc, $query0);
```

```
while ($row0 = mysqli_fetch_assoc($result0))
{
   $doors_id = $row0['doors_id'];
   $doors_id_arr = array();
   $doors_id_arr[(integer)$doors_id] = 'ren_name';
   $_SESSION['doors_id_arr'] = $doors_id_arr;

   if(!missing_engine())
   {
      $fleet_doors_id_arr[] = $row0['doors_id'];
   }
}

foreach($fleet_doors_id_arr as $key => $doors_id)
{
   for ($run=0; $run<2; $run++)
   {
      mysqli_query ($dbc4, "TRUNCATE TABLE osm_vehicle");
      include("initialize_fleet.php");
      include("initialize_year.php");
      include("create_fleet.php");
      include("calculate_early_credit.php");
      include("insert_vehicle_average_age.php");
      include("get_fleet_info.php");
//echo $doors_id . ' hp ' . $total_hp . 'oit<BR>';
      if($total_hp>0)
      {
//echo $doors_id . ' hp ' . $total_hp . 'in<BR>';
         include("get_fleet_target.php");
         include("get_fleet_average.php");
         if($run==0) include("initialize_base.php");
         if($run==1) include("initialize_scenario.php");
/*
         for ($calendar_year=2011; $calendar_year < $end_year+1; $calendar_year++)
         {
            include("initialize_year.php");

            if (isset($_POST['run_sim1']) and $calendar_year==2015)
            {
               $turnover_carryover =+ $retirement_2006_2010_credit/2;
            }

            include("create_new_calendar_year.php");
            include("insert_vehicle_average_age.php");
            include("get_fleet_info.php");
            include("get_fleet_target.php");
```

```
            include("get_fleet_average.php");
            include("set_nox_pm_compliance_flags.php");

            if($run == 0
              or ($run == 1 and ($fleet_target_nox == 0
                           or $turnover_rate_forced == 0
                           or $meets_fleet_target_nox)))
            {
               include("initialize_base.php");
            }
            else
            {
               include("initialize_scenario.php");
            }

            include("do_turnover_base.php");

            if($run == 1
               and $fleet_target_nox > 0
               and !$meets_fleet_target_nox
               and $turnover_rate_forced > 0
               and !$free_2014)
            {
               include("set_turnover_rate_and_veh_pool.php");
               include("do_turnover.php");
            }

             if($run == 1
               and $fleet_target_pm > 0
               and !$meets_fleet_target_pm
               and !$turnover_only
               and !$free_2014)
            {
               include("set_retrofit_rate_and_veh_pool.php");
               include("do_retrofit.php");
            }
        }
      */    include("move_fleet_to_base_scenario.php");
       }
    }
}

?>
```

```
<?php  #set_nox_pm_compliance_flags
```

```
if ($run==0)
{
   $meets_fleet_target_nox = 1;
   $meets_fleet_target_pm = 1;
}
else
{
   if ($fleet_average_nox <= $fleet_target_nox)
   {
      $meets_fleet_target_nox = 1;
   }
   else
   {
      $meets_fleet_target_nox = 0;
   }

   if ($fleet_average_pm <= $fleet_target_pm)
   {
      $meets_fleet_target_pm = 1;
   }
   else
   {
      $meets_fleet_target_pm = 0;
   }
}

?>
```

```
<?php #set_retrofit_rate_and_veh_pool


if($calendar_year == 2020)
{
   $retrofit_carryover = 0;
}

$retrofit_forced = $total_hp * $retrofit_rate_forced;

$retrofit_carryover += $turnover_tier4 + $turnover_tier;

if($run==1
   and $retrofit_forced > 0
   and !$meets_fleet_target_pm)
{
```

```
    if($expiring_pm_credit >= $retrofit_forced)
    {
        $expiring_pm_credit = $expiring_pm_credit - $retrofit_forced;
        $retrofit_needed = 0;
    }
    else
    {
        $retrofit_needed = $retrofit_forced - $expiring_pm_credit;
        $expiring_pm_credit = 0;
        if($retrofit_carryover >= $retrofit_needed)
        {
            $retrofit_carryover = $retrofit_carryover - $retrofit_needed;
            $retrofit_needed = 0;
        }
        else
        {
            $retrofit_needed = $retrofit_needed - $retrofit_carryover;
            $retrofit_carryover = 0;
        }
    }
}
else
{
            $retrofit_needed = 0;
}

$query = "SELECT ein, eng_hp
          FROM osm_vehicle
          WHERE calendar_year = '$calendar_year'
            AND ('$calendar_year' - eng_model_year > 4)
            AND (vdecs_pm = 1)
            AND (eng_index_tier < 'T4')
            AND eng_model_year > 1991
            AND eng_hp >= 56
            AND eng_hp <= 500
          ORDER BY eng_index_tier DESC, eng_hp DESC, ein";

if(!$result = @mysqli_query ($dbc4, $query)) $message .= '_260';

?>
```

```
<?php # set_turnover_rate_and_veh_pool


$turnover_forced = $total_hp * $turnover_rate_forced;
```

```
$tunover_forced_minus_base = $turnover_forced - $turnover_actual_base;

if($run==1
   and $tunover_forced_minus_base > 0
   and !$meets_fleet_target_nox)
{
  if($expiring_nox_credit >= $tunover_forced_minus_base)
   {
      $expiring_nox_credit = $expiring_nox_credit - $tunover_forced_minus_base;
      $turnover_needed = 0;
   }
   else
   {
      $turnover_needed = $tunover_forced_minus_base - $expiring_nox_credit;
      $expiring_nox_credit = 0;

      if($turnover_carryover >= $turnover_needed)
      {
         $turnover_carryover = $turnover_carryover - $turnover_needed;
         $turnover_needed = 0;
      }
      else
      {
         $turnover_needed = $turnover_needed - $turnover_carryover;
         $turnover_carryover = 0;
      }
   }
}
else
{
   $turnover_needed = 0;
}

if($calendar_year < 2013)
{
   $query_fleet = "SELECT *
               FROM osm_vehicle
               WHERE doors_id = '$doors_id'
                 AND calendar_year = '$calendar_year'
                 AND (calendar_year - veh_model_year) >=
$new_vehicle_turnover_exemption_age
                 AND eng_index_tier < 'T1U'
                 AND '$calendar_year' - calendar_year_retrofit >= 6
                 AND low_use = 0
               ORDER BY eng_index_tier, veh_relative_age DESC, ein";
```

```
}
else
{
   $query_fleet = "SELECT *
                FROM osm_vehicle
                WHERE doors_id = '$doors_id'
                  AND calendar_year = '$calendar_year'
                  AND (calendar_year - veh_model_year) >=
$new_vehicle_turnover_exemption_age
                  AND eng_index_tier < 'T4I'
                  AND '$calendar_year' - calendar_year_retrofit >= 6
                  AND low_use = 0
                ORDER BY eng_index_tier, veh_relative_age DESC, ein";
}

?>
```

```
<!-- show_costs_total -->

<TABLE border="1" cellpadding="5" cellspacing="5" bgcolor="#CEEFBD">


   <TR><TD bgcolor="#E0E0E0">Run</TD>
      <TD bgcolor="#E0E0E0">Calendar Year</TD>
      <TD bgcolor="#E0E0E0">Full Turnover</TD>
      <TD bgcolor="#E0E0E0">Retrofit</TD>
      <TD bgcolor="#E0E0E0">Full Annual</TD>
      <TD bgcolor="#E0E0E0">PV Full Annual</TD>
   </TR>

<?php

   $query = "SELECT calendar_year,
                  SUM(cost_annual_full_turnover) AS cost_annual_full_turnover,
                  SUM(cost_annual_retrofit) AS cost_annual_retrofit,
                  SUM(cost_annual_full) AS cost_annual_full,
                  SUM(cost_annual_full_pv) AS cost_annual_full_pv
          FROM osm_fleet_base_costs_cumulative
          GROUP BY calendar_year
          ORDER BY calendar_year";

$result = @mysqli_query ($dbc4, $query);

if(mysqli_num_rows($result)==0)
```

```
{
   echo '<TR><TD colspan="13" bgcolor="#FF0000"><CENTER><B>
              No rows in cost tables</TD></TR>';
}

while ($row = mysqli_fetch_array($result))
{
   $cost_annual_full_turnover = number_format($row['cost_annual_full_turnover'], 0,
"",",");
   $cost_annual_retrofit      = number_format($row['cost_annual_retrofit'], 0, "",",");
   $cost_annual_full          = number_format($row['cost_annual_full'], 0, "",",");
   $cost_annual_full_pv       = number_format($row['cost_annual_full_pv'], 0, "",",");

   echo '<TR>';
   echo '<TD bgcolor="#FFFFFF"> Base </TD>';
   echo '<TD bgcolor="#FFFFFF">' . $row['calendar_year'] . '</TD>';
   echo '<TD bgcolor="#FFFFC6">' . $cost_annual_full_turnover . '</TD>';
   echo '<TD bgcolor="#FFFFC6">' . $cost_annual_retrofit . '</TD>';
   echo '<TD bgcolor="#FFFFC6">' . $cost_annual_full . '</TD>';
   echo '<TD bgcolor="#FFFFC6">' . $cost_annual_full_pv . '</TD>';
   echo '</TR>';
}

echo '<TR><TD bgcolor="#E0E0E0">Run</TD>
      <TD bgcolor="#E0E0E0">Calendar Year</TD>
      <TD bgcolor="#E0E0E0">Full Turnover</TD>
      <TD bgcolor="#E0E0E0">Retrofit</TD>
      <TD bgcolor="#E0E0E0">Full Annual</TD>
      <TD bgcolor="#E0E0E0">PV Full Annual</TD>
     </TR>';

   $query = "SELECT calendar_year,
                 SUM(cost_annual_full_turnover) AS cost_annual_full_turnover,
                 SUM(cost_annual_retrofit) AS cost_annual_retrofit,
                 SUM(cost_annual_full) AS cost_annual_full,
                 SUM(cost_annual_full_pv) AS cost_annual_full_pv
           FROM osm_fleet_scenario_costs_cumulative
           GROUP BY calendar_year
           ORDER BY calendar_year";
$result = @mysqli_query ($dbc4, $query);

if(mysqli_num_rows($result)==0)
{
   echo '<TR><TD colspan="13" bgcolor="#FF0000"><CENTER><B>
              No rows in cost tables</TD></TR>';
}
```

```
while ($row = mysqli_fetch_array($result))
{
   $cost_annual_full_turnover = number_format($row['cost_annual_full_turnover'], 0,
"",",");
   $cost_annual_retrofit      = number_format($row['cost_annual_retrofit'], 0, "",",");
   $cost_annual_full          = number_format($row['cost_annual_full'], 0, "",",");
   $cost_annual_full_pv       = number_format($row['cost_annual_full_pv'], 0, "",",");

   echo '<TR>';
   echo '<TD bgcolor="#FFFFFF"> Scenario </TD>';
   echo '<TD bgcolor="#FFFFFF">' . $row['calendar_year'] . '</TD>';
   echo '<TD bgcolor="#FFFFC6">' . $cost_annual_full_turnover . '</TD>';
   echo '<TD bgcolor="#FFFFC6">' . $cost_annual_retrofit . '</TD>';
   echo '<TD bgcolor="#FFFFC6">' . $cost_annual_full . '</TD>';
   echo '<TD bgcolor="#FFFFC6">' . $cost_annual_full_pv . '</TD>';
   echo '</TR>';
}

?>

</TABLE><BR><BR>
```

IV. Off-Road Emission Inventory Model

a. Off-road Emissions Inventory Model and Input Tables

The Off-road Emissions Inventory model uses the final output generated by the Off-road Simulation Model (OSM – see previous section for model details). The two sub-procedures titled in this printout as ‘OSM Emissions’ and ‘Emissions Coefficients’ encompass the programming behind the emissions inventory. This code runs in Microsoft Access and relies on input tables that were developed by ARB staff. ‘Emissions Coefficients’ is run separately from ‘OSM Emissions’ and is explained within that sub-procedure. The ‘OSM Emissions’ procedure mainly handles formatting and emissions manipulation. The comments that accompany each code are provided to help users understand the purpose of the code (**comments/explanations are in non-italicized font**). This printout is identical to the code within the Emissions Inventory model provided at this time.

Input Tables

The following list contains all the input tables. Each table has the following information: the name of the table, the name of the section(s) it is located, how it is used, and the fields and units used in that section:

Activity / Cumulative Hours Table – “ActivityCmHrs”

EmissionsCoefficients procedure:

- Section: “EmissionFactors”
  Field name: “Cumulative Hours Final”
  Purpose: Cumulative hours for individual vehicle types relate the total hours a vehicle type has been used in its life to the deterioration rate, i.e. emission rates are slightly higher with higher cumulative hours
  Units: hours
- Section: “LoadActivity”
  Field name: “DOORS Adjusted Final”
  Purpose: Annual hours for individual vehicle types are stored in a field to be multiplied against emission rates to get annual emissions (hours are renormalized in osmEmissions before being used, see procedures)
  Units: hours/year

Air Basin Allocation Table – “AirBasinAllocation”

osmEmissions procedure:

- Sections: “BaselineAllocation2*”* and “ScenarioAllocation2”
  Field name: “Allocation”
  Purpose: Allocation factors distribute aggregate emissions into air basins, both sections have identical methodologies
  Units: unitless

Population Adjustment Table – “BasePop”

osmEmissions procedure:

- Section: “AdjustPop1”
  Field name: “Adjusted Population”
  Purpose: Populations input to the *osmEmissions* procedure are incomplete and so the estimated real-world population found in BasePop is used to scale up population to reflect the accurate population
  Units: number of vehicles

Brake Specific Fuel Consumption Table – “BSFuelConsumption”

EmissionsCoefficients procedure:

- Section: “BSFCFactors”
  Field name: “BSFC”
  Purpose: Fuel Consumption rates are multiplied to calculate total fuel consumption
  Units: pounds of fuel / (horsepower * hour)

Emission Factors / Deterioration Rate Table – “EMFACupdate”

EmissionsCoefficients procedure:

- Section: “EmissionFactors”
  Field names: NOx, NOx Det, PM, PM Det, HC, HC Det
  Purpose: Emission factors (NOx, PM, HC) are used to calculate total emissions. Deterioration rates (NOx Det, PM Det, HC Det) are used with cumulative hours (see above) to calculate the increase in emissions due to a vehicle’s age
  Units (EF): grams / (horsepower * hour)
  Units (DR): grams / (horsepower * hour * hour)

Fuel Correction Factor Table – “FuelCorrectionFactorUpdate”

EmissionsCoefficients procedure:

- Section: “FuelCorrectionFactor”
  Field name: NOX, PM, HC
  Purpose: Factors are multiplied against emissions to account for the reductions from fuel improvements
  Units: unitless

Growth Factors – “GrowthFactor”

osmEmissions procedure:

- Section: “GrowEmissions”
  Field name: “GrowthFactor”
  Purpose: Growth factors are multiplied against rated horsepower to reflect the economic growth expected within different sectors
  Units: unitless

Load Factors – “LoadFactor”

EmissionsCoefficients procedure:

- Section: “LoadFactor”
  Field name: “Adj ARB LF”

Purpose: Load factors are multiplied against emissions to reflect the actual running conditions during operations.
Units: unitless

Lookup Equipment Type Table – “LookupEquipmentType”
EmissionsCoefficients and *osmEmissions* procedures:

- Used throughout model
  Field names: “Equipment Type” and “EquipmentTypeID”
  Purpose: To save space, equipment IDs are used instead of the actual names of equipment types

b. Off-road Emissions Inventory Code

Terminology

Some terminology used in the Off-road Emissions Inventory code is used for simplicity. The following terms are defined for this code only – other models may use different naming conventions and these do not represent the any official ARB definitions.

Baseline (sometimes ‘base’): the OSM output generated under business-as-usual
Scenario (sometimes ‘scen’): the OSM output generated under the proposed regulation
Emission Coefficient: a number that contains load factor, fuel correction factor, emission factor information all in one. It can be multiplied against a population and horsepower to generate emissions.
‘Unique vehicle type’ and ‘specificity’: There are different vehicle types such as ‘forklifts’ or ‘excavators’ so in common terms a specific vehicle type could mean what type of vehicle it is. However, in this model vehicles are considered unique or different from each other if they have different ages, horse-powers, or at times fleet location. Throughout the code the ‘level of specificity’ refers to how a particular section is considering differences (i.e. at times vehicles are unique just by name, at other times they are specific or unique to age, horsepower, etc.).

EMISSIONS COEFFICIENTS

Sub EmCo()

DoCmd.SetWarnings False

'The 'EmissionsCoefficients' table is created as a generic multiplier for any population of vehicles. The table contains information about emission factors, fuel correction factors, load factors, fuel consumption rates, and activities for all unique types of equipment. A piece of equipment is designated unique by having a combination of calendar year, model year, horsepower, and equipment type. Equipment type categories have been designated to reflect general off-road equipment categories. Each record in EmissionsCoefficients is to be used as a multiplier for calculating emissions. Units are grams / (horsepower * hour). Activity is loaded as a separate value so that renormalization issues can be worked out in a separate module without affecting the emissions coefficients themselves.

'MakeUniqueVehicles - The following query is used to create a table with all unique categories. The table 'GrowthFactor' contains all calendar years; 'EMFACupdate' contains all horsepowers and model years; 'LookupEquipmentType' contains all equipment types. No values are entered or calculated and a placeholder of '1' is used for each emission category for each unique equipment category. Calendar Years are limited from 2009 to 2030 and model years from 89 years before a calendar year to 1 year after. These limitations are to include extreme outliers and allow for any slight future variations while making results uniform (same number of MYs in each CY). Equipment type IDs are used instead of equipment names for the sake of saving memory space and will be used consistently throughout the program.
'INTO EmissionsCoefficients

DoCmd.RunSQL "SELECT GrowthFactor.CalendarYear, LookupEquipmentType.EquipmentTypeID, EMFACupdate!HP AS HorsePower, EMFACupdate![Model Year] AS ModelYear, [GrowthFactor]![CalendarYear]-[EMFACupdate]![Model Year] AS Age, CDbl(1) AS BSFC, CDbl(1) AS HC, CDbl(1) AS NOx, CDbl(1) AS PM INTO EmissionsCoefficients FROM EMFACupdate, GrowthFactor INNER JOIN LookupEquipmentType ON GrowthFactor.EquipmentClass = LookupEquipmentType.[Equipment Class] WHERE (((GrowthFactor.CalendarYear) <= 2029 And (GrowthFactor.CalendarYear) >= 2009) And (([EMFACupdate]![Model Year]) >= ([GrowthFactor]![CalendarYear] - 89) And ([EMFACupdate]![Model Year]) <= ([GrowthFactor]![CalendarYear] + 1) And ([EMFACupdate]![Model Year]) <= 2030)) ORDER BY GrowthFactor.CalendarYear, LookupEquipmentType.EquipmentTypeID, EMFACupdate!HP, EMFACupdate![Model Year] DESC;"

'EmissionFactors - Cumulative hours from table 'ActivityCmHrs' and emission factors and deterioration rates from table 'EMFACupdate' are used to calculate emissions factor values for EmissionCoefficients. The following equations all calculate a grams/(HP*hour) emission factor. Deterioration rates and emission factors are derived to multiply into these equations as shown. They are multiplied against the placeholders which are initially set at 1. The table 'LookupEquipmentType' is used to join EmissionsCoefficients to ActivityCmHrs as ID's are used in large tables and names are used in small ones to identify vehicle type (this is a space saving technique). BSFC rates are calculated separately in the next query.
'NOx = 1 * ([Emission Factor] + ([Deterioration Rate] * [Cumulative Hours]))
'PM = 1 * ([Emission Factor] + ([Deterioration Rate] * [Cumulative Hours]))
'HC = 1 * ([Emission Factor] + ([Deterioration Rate] * [Cumulative Hours]))
'UPDATE EmissionsCoefficients

DoCmd.RunSQL "UPDATE ActivityCmHrs INNER JOIN ((EmissionsCoefficients INNER JOIN EMFACupdate ON (EmissionsCoefficients.HorsePower =

```
EMFACupdate.HP) AND (EmissionsCoefficients.ModelYear =
EMFACupdate.[Model Year])) INNER JOIN LookupEquipmentType ON
EmissionsCoefficients.EquipmentTypeID =
LookupEquipmentType.EquipmentTypeID) ON (ActivityCmHrs.Equipment =
LookupEquipmentType.[Equipment Type]) AND (ActivityCmHrs.Age =
EmissionsCoefficients.Age) SET EmissionsCoefficients.HC =
[EmissionsCoefficients]![HC]*([EMFACupdate]![HC]+([EMFACupdate]![HC
Det]*[ActivityCmHrs]![Cumulative Hours Final])), EmissionsCoefficients.NOx =
[EmissionsCoefficients]![NOx]*([EMFACupdate]![NOx]+([EMFACupdate]![NOx
Det]*[ActivityCmHrs]![Cumulative Hours Final])), EmissionsCoefficients.PM =
[EmissionsCoefficients]![PM]*([EMFACupdate]![PM]+([EMFACupdate]![PM
Det]*[ActivityCmHrs]![Cumulative Hours Final]));"
```

'BSFCFactors - Brake specific fuel consumption rates from the table 'BSFuelConsumption' are multiplied directly as in the following equation to the placeholder '1'. The values calculated have units of pounds/(HP*hour). BSFC values are specific to the model year and horsepower of a given vehicle. However, when load factors are applied to the EmissionsCoefficients table, BSFC becomes vehicle type specific as well.
'BSFC = BSFC * 1
'UPDATE EmissionsCoefficients

```
DoCmd.RunSQL "UPDATE BSFuelConsumption INNER JOIN
EmissionsCoefficients ON (BSFuelConsumption.MY =
EmissionsCoefficients.ModelYear) AND (BSFuelConsumption.[ARB HP] =
EmissionsCoefficients.HorsePower) SET EmissionsCoefficients.BSFC =
EmissionsCoefficients!BSFC*BSFuelConsumption!BSFC;"
```

'LoadFactor - Load factors from table 'LoadFactor' (field: 'Adj ARB LF') are unitless and are multiplied directly to EmissionsCoefficients. They are specific to equipment type only. They apply to all four coefficients: BSFC, NOx, PM, and HC.
'UPDATE EmissionsCoefficients

```
DoCmd.RunSQL "UPDATE LoadFactor INNER JOIN EmissionsCoefficients ON
LoadFactor.EquipmentTypeID = EmissionsCoefficients.EquipmentTypeID SET
EmissionsCoefficients.BSFC = [EmissionsCoefficients]![BSFC]*[LoadFactor]![Adj
ARB LF], EmissionsCoefficients.HC =
[EmissionsCoefficients]![HC]*[LoadFactor]![Adj ARB LF],
EmissionsCoefficients.NOx = [EmissionsCoefficients]![NOx]*[LoadFactor]![Adj
ARB LF], EmissionsCoefficients.PM =
[EmissionsCoefficients]![PM]*[LoadFactor]![Adj ARB LF];"
```

'FuelCorrectionFactor - Fuel Correction Factors from the table 'FuelCorrectionFactorUpdate' are unitless and are multiplied directly to NOx and PM but not HC or BSFC. They are model year and horsepower specific. BSFC has no FCF and

HC has a value of 0.72 that applies to all horsepower and model year categories. This value is coded into the following query.
'UPDATE EmissionsCoefficients

> *DoCmd.RunSQL "UPDATE EmissionsCoefficients INNER JOIN FuelCorrectionFactorUpdate ON (EmissionsCoefficients.ModelYear = FuelCorrectionFactorUpdate.[Model Yr]) AND (EmissionsCoefficients.HorsePower = FuelCorrectionFactorUpdate.Hp) SET EmissionsCoefficients.NOx = [EmissionsCoefficients]![NOx]*[FuelCorrectionFactorUpdate]![NOX], EmissionsCoefficients.PM = [EmissionsCoefficients]![PM]*[FuelCorrectionFactorUpdate]![PM], EmissionsCoefficients.HC = [EmissionsCoefficients]![HC]*0.72;"*

'ActivityColumn - Activity must be renormalized in the module ‘OsmEmissions’ before it can be applied to equipment or emissions. This query creates a column for activity so that it can be manipulated separately from emissions coefficients values during renormalization (see OsmEmissions).
'ALTER EmissionsCoefficients

> *DoCmd.RunSQL "ALTER TABLE EmissionsCoefficients ADD COLUMN [Activity] DOUBLE;"*

'LoadActivity - The following query uploads activity values in units of hours/year by equipment type and age into the blank column created above. These are taken from the table 'ActivityCmHrs'. The table 'LookupEquipmentType' is used again for linking equipment ID's to the corresponding equipment type names. These will be renormalized in the module ‘osmEmissions’ before they are multiplied to emissions and populations.
'UPDATE EmissionsCoefficients

> *DoCmd.RunSQL "UPDATE ActivityCmHrs INNER JOIN (EmissionsCoefficients INNER JOIN LookupEquipmentType ON EmissionsCoefficients.EquipmentTypeID = LookupEquipmentType.EquipmentTypeID) ON (ActivityCmHrs.Age = EmissionsCoefficients.Age) AND (ActivityCmHrs.Equipment = LookupEquipmentType.[Equipment Type]) SET EmissionsCoefficients.Activity = [ActivityCmHrs]![DOORS Adjusted Final];"*

> *DoCmd.SetWarnings True*

End Sub

OSM EMISSIONS

Sub OsmEmissions()

DoCmd.SetWarnings False

'LoadTemplate - OSM output is loaded 'as is' with designated names 'Osm_vehicle_base' and 'Osm_vehicle_scenario'. The table 'TemplateOSM' is a table for the emissions inventory that is used for formatting OSM output. The following query creates a copy of TemplateOSM for inputting both OSM tables. A copy is created so that each time this routine is run the template for formatting stays untouched.
'INTO osmEmissions

DoCmd.RunSQL "SELECT TemplateOSM. INTO osmEmissions FROM TemplateOSM;"*

'InputOSM - Input all needed information from OSM output tables 'Osm_vehicle_base' and 'Osm_vehicle_scenario.' These two tables are condensed to one for the sake of saving memory space and ease of programming. Tables share common characteristics like calendar years, horsepower, fleet ID, etc., so that no informational integrity is lost in condensing to one table. Instead of loading equipment type names, the table LookupEquipmentType is used to load ID numbers for all the unique vehicle types. This is also done for the sake of saving memory space and programming ease. From here on equipment ID's are used as much as possible for consistency.
'APPEND osmEmissions

```
DoCmd.RunSQL "INSERT INTO osmEmissions ( FleetSize, DoorsID,
CalendarYear, EquipmentTypeID, HP, HorsepowerBin, LowUse, BaseAge,
BaseModelYear, BaseTier, BaseCYRetrofit, BaseVDECSNOx, BaseVDECSPM,
ScenAge, ScenModelYear, ScenTier, ScenCYRetrofit, ScenVDECSNOx,
ScenVDECSPM ) SELECT osm_vehicle_base.Field1, osm_vehicle_base.Field2,
osm_vehicle_base.Field4, LookupEquipmentType.EquipmentTypeID,
osm_vehicle_base.Field8, osm_vehicle_base.Field9, osm_vehicle_base.Field10,
osm_vehicle_base!Field4-osm_vehicle_base!Field7 AS BaseAge,
osm_vehicle_base.Field7, osm_vehicle_base.Field17,
osm_vehicle_base.Field12, osm_vehicle_base.Field13,
osm_vehicle_base.Field14, osm_vehicle_base!Field4-
osm_vehicle_scenario!Field7 AS ScenAge, osm_vehicle_scenario.Field7,
osm_vehicle_scenario.Field17, osm_vehicle_scenario.Field12,
osm_vehicle_scenario.Field13, osm_vehicle_scenario.Field14 " & _
"FROM (osm_vehicle_base INNER JOIN LookupEquipmentType ON
osm_vehicle_base.Field5 = LookupEquipmentType.[Equipment Type]) INNER
JOIN osm_vehicle_scenario ON (osm_vehicle_base.Field2 =
osm_vehicle_scenario.Field2) AND (osm_vehicle_base.Field8 =
osm_vehicle_scenario.Field8) AND (osm_vehicle_base.Field4 =
osm_vehicle_scenario.Field4) AND (osm_vehicle_base.Field3 =
osm_vehicle_scenario.Field3) AND (osm_vehicle_base.Field1 =
osm_vehicle_scenario.Field1) AND (LookupEquipmentType.[Equipment Type] =
osm_vehicle_scenario.Field5);"
```

'EmissionsActivityColumns - Activity will be loaded and emissions calculated from the table 'EmissionsCoefficients.' The following query simply creates blank columns for these fields.
'ALTER osmEmissions

DoCmd.RunSQL "ALTER TABLE osmEmissions ADD COLUMN [BaseActivity] DOUBLE, [BaseBSFC] DOUBLE, [BaseNOx] DOUBLE, [BasePM] DOUBLE, [BaseHC] DOUBLE, [ScenActivity] DOUBLE, [ScenBSFC] DOUBLE, [ScenNOx] DOUBLE, [ScenPM] DOUBLE, [ScenHC] DOUBLE;"

'Some of the following procedures use the table 'EmissionsCoefficients.' This table is created by the module 'EmCoefficient', sub routine 'EmCo'. The code and explanation for deriving the table can be found there.

'LoadBaseActivity - The following query loads activity for baseline emissions from the table 'EmissionsCoefficients'. Equipment type and model year designate the activity in hours per year. Baseline and Scenario activity are loaded separately because the same piece of equipment can have different ages under each scenario.
'UPDATE osmEmissions

DoCmd.RunSQL "UPDATE osmEmissions INNER JOIN EmissionsCoefficients ON (osmEmissions.BaseModelYear = EmissionsCoefficients.ModelYear) AND (osmEmissions.HorsepowerBin = EmissionsCoefficients.HorsePower) AND (osmEmissions.CalendarYear = EmissionsCoefficients.CalendarYear) AND (osmEmissions.EquipmentTypeID = EmissionsCoefficients.EquipmentTypeID) SET osmEmissions.BaseActivity = IIf([osmEmissions]![LowUse]=0,[EmissionsCoefficients]![Activity],200);"

'LoadScenActivity - Loads scenario activity based on equipment type and the scenario's age. Same as LoadBaseActivity using scenario input.
'UPDATE osmEmissions

DoCmd.RunSQL "UPDATE osmEmissions INNER JOIN EmissionsCoefficients ON (osmEmissions.ScenModelYear = EmissionsCoefficients.ModelYear) AND (osmEmissions.HorsepowerBin = EmissionsCoefficients.HorsePower) AND (osmEmissions.CalendarYear = EmissionsCoefficients.CalendarYear) AND (osmEmissions.EquipmentTypeID = EmissionsCoefficients.EquipmentTypeID) SET osmEmissions.ScenActivity = IIf([osmEmissions]![LowUse]=0,[EmissionsCoefficients]![Activity],200);"

''''''The following set of queries is to renormalize the fleet specific yearly activity. Growth is handled separately from activity so increases in activity resulting from vehicle turnover are renormalized so total fleet activity stays constant. Specificity within a fleet: groupings by fleet, equipment type, and horsepower are renormalized.

'ActivityRenormalize1 - The first step is summing the total number of hours each calendar year specific to fleet, equipment type, and horsepower. These activity values vary by calendar year and must be renormalized. Base and scenario activity are summed together.

'INTO Activity1

```
DoCmd.RunSQL "SELECT osmEmissions.CalendarYear,
osmEmissions.DoorsID, osmEmissions.EquipmentTypeID,
osmEmissions.HorsepowerBin, Sum(osmEmissions.BaseActivity) AS
SumOfBaseActivity, Sum(osmEmissions.ScenActivity) AS SumOfScenActivity
INTO Activity1 FROM osmEmissions GROUP BY osmEmissions.CalendarYear,
osmEmissions.DoorsID, osmEmissions.EquipmentTypeID,
osmEmissions.HorsepowerBin, osmEmissions.LowUse HAVING
(((osmEmissions.LowUse)<>1));"
```

'ActivityRenormalize2 - The summation of activity just as in ActivityRenormalize1 exclusive to calendar year 2009. This is used as the numerator for the fraction calculated in ActivityRenormalize3

'INTO Activity2

```
DoCmd.RunSQL "SELECT osmEmissions.CalendarYear,
osmEmissions.DoorsID, osmEmissions.EquipmentTypeID,
osmEmissions.HorsepowerBin, Sum(osmEmissions.BaseActivity) AS
SumOfBaseActivity, Sum(osmEmissions.ScenActivity) AS SumOfScenActivity
INTO Activity2 FROM osmEmissions GROUP BY osmEmissions.CalendarYear,
osmEmissions.DoorsID, osmEmissions.EquipmentTypeID,
osmEmissions.HorsepowerBin, osmEmissions.LowUse HAVING
(((osmEmissions.CalendarYear)=2009) AND ((osmEmissions.LowUse)<>1));"
```

'ActivityRenormalize3 - Within a given fleet, equipment type, and horsepower (just as Activity1 and Activity2), the base year activity (2009) is divided by each successive calendar year's activity. The fraction activity2009/activityCY ('CY' representing any given calendar year) is to be multiplied against every individual vehicle cancelling out the overall fleet activity growth or reduction relative to 2009 thus recovering the original overall activity.

'INTO Activity3

```
DoCmd.RunSQL "SELECT Activity1.CalendarYear, Activity1.DoorsID,
Activity1.EquipmentTypeID, Activity1.HorsepowerBin,
[Activity2]![SumOfBaseActivity]/[Activity1]![SumOfBaseActivity] AS
RenormFactorBase,
[Activity2]![SumOfScenActivity]/[Activity1]![SumOfScenActivity] AS
RenormFactorScen INTO Activity3 FROM Activity1 INNER JOIN Activity2 ON
(Activity1.HorsepowerBin = Activity2.HorsepowerBin) AND
(Activity1.EquipmentTypeID = Activity2.EquipmentTypeID) AND
(Activity1.DoorsID = Activity2.DoorsID);"
```

''''''''

'RenormalizeActivity - Renormalization Fractions are multiplied directly to every individual equipment in osmEmissions.
'UPDATE osmEmissions

DoCmd.RunSQL "UPDATE Activity3 INNER JOIN osmEmissions ON (Activity3.CalendarYear = osmEmissions.CalendarYear) AND (Activity3.DoorsID = osmEmissions.DoorsID) AND (Activity3.EquipmentTypeID = osmEmissions.EquipmentTypeID) AND (Activity3.HorsepowerBin = osmEmissions.HorsepowerBin) SET osmEmissions.BaseActivity = IIf([osmEmissions]![LowUse]=0,[osmEmissions]![BaseActivity][Activity3]![RenormFactorBase],[osmEmissions]![BaseActivity]), osmEmissions.ScenActivity = IIf([osmEmissions]![LowUse]=0,[osmEmissions]![ScenActivity]*[Activity3]![RenormFactorScen],[osmEmissions]![ScenActivity]);"*

'BaselineEmissions - Values from 'EmissionsCoefficients' are used to calculate brake specific fuel consumption (BSFC), NOx Emissions, PM emissions, and HC emissions. PM and NOx have VDECS factors representing filter reductions in emissions that are multiplied directly to results. EmissionsCoefficients are all units of g/(HP*hr) except BSFC which is pounds/(HP*hr). These units allow the direct multiplication against the activity and horsepower in osmEmissions. For information on EmissionsCoefficients see module 'EmCoefficient'
'BSFC = BSFC*Activity*HP, NOx = NOx*Activity*HP*VDECSNOx, PM = PM*Activity*HP*VDECSPM, HC = HC*Activity*HP
'UPDATE osmEmissions

DoCmd.RunSQL "UPDATE EmissionsCoefficients INNER JOIN osmEmissions ON (EmissionsCoefficients.EquipmentTypeID = osmEmissions.EquipmentTypeID) AND (EmissionsCoefficients.CalendarYear = osmEmissions.CalendarYear) AND (EmissionsCoefficients.HorsePower = osmEmissions.HorsepowerBin) AND (EmissionsCoefficients.ModelYear = osmEmissions.BaseModelYear) SET osmEmissions.BaseBSFC = [osmEmissions]![BaseActivity][EmissionsCoefficients]![BSFC]*[osmEmissions]![HP], osmEmissions.BaseNOx = [osmEmissions]![BaseActivity]*[EmissionsCoefficients]![NOx]*[osmEmissions]![BaseVDECSNOx]*[osmEmissions]![HP], osmEmissions.BasePM = [osmEmissions]![BaseActivity]*[EmissionsCoefficients]![PM]*[osmEmissions]![HP]*[osmEmissions]![BaseVDECSPM], osmEmissions.BaseHC = [osmEmissions]![BaseActivity]*[EmissionsCoefficients]![HC]*[osmEmissions]![HP];"*

'ScenarioEmissions - Same methodology as 'BaselineEmissions' used to calculate scenario emissions. Scenario model year and age are used instead of baseline values.
'UPDATE osmEmissions

DoCmd.RunSQL "UPDATE EmissionsCoefficients INNER JOIN osmEmissions ON (EmissionsCoefficients.EquipmentTypeID = osmEmissions.EquipmentTypeID) AND (EmissionsCoefficients.CalendarYear = osmEmissions.CalendarYear) AND (EmissionsCoefficients.HorsePower = osmEmissions.HorsepowerBin) AND (EmissionsCoefficients.ModelYear = osmEmissions.ScenModelYear) SET osmEmissions.ScenBSFC = [osmEmissions]![ScenActivity][EmissionsCoefficients]![BSFC]*[osmEmissions]![HP], osmEmissions.ScenNOx = [osmEmissions]![ScenActivity]*[EmissionsCoefficients]![NOx]*[osmEmissions]![ScenVDECSNOx]*[osmEmissions]![HP], osmEmissions.ScenPM = [osmEmissions]![ScenActivity]*[EmissionsCoefficients]![PM]*[osmEmissions]![ScenVDECSPM]*[osmEmissions]![HP], osmEmissions.ScenHC = [osmEmissions]![ScenActivity]*[EmissionsCoefficients]![HC]*[osmEmissions]![HP];"*

""""The following set of queries calculates the adjustment factor that reflects the differences between reported equipment available in osmEmissions and the actual population. The table 'BasePop' reflects real-world populations and is used to calculate the adjustment factor.

'INTO SumPopET1 - The population in osmEmissions Calendar Year 2009 is counted.

DoCmd.RunSQL "SELECT osmEmissions.CalendarYear, osmEmissions.EquipmentTypeID, Count(osmEmissions.CalendarYear) AS CountOfEIN INTO SumPopET1 FROM osmEmissions GROUP BY osmEmissions.CalendarYear, osmEmissions.EquipmentTypeID HAVING (((osmEmissions.CalendarYear)=2009));"

'INTO SumPopET2 - 'BasePopulation' contains a real-world approximation of populations. A common adjustment factor was applied across every equipment type resulting in the 'Adjusted Population'. The proportions of these populations relative to the osmEmissions populations are found in the following query. These factors will scale up populations to a more real-world value. Factors are specific to equipment type.

DoCmd.RunSQL "SELECT LookupEquipmentType.EquipmentTypeID, [BasePop]![Adjusted Population]/[SumPopET1]![CountOfEIN] AS AdjustmentFraction INTO SumPopET2 FROM BasePop INNER JOIN (SumPopET1 INNER JOIN LookupEquipmentType ON SumPopET1.EquipmentTypeID = LookupEquipmentType.EquipmentTypeID) ON BasePop.[Equiment Type] = LookupEquipmentType.[Equipment Type];"

'PopulationAdjustment - Population adjustment factor calculated above is used to adjust emissions.
'BSFC = BSFC*PopFactor, NOx = NOx*PopFactor, PM = PM*PopFactor, HC = HC*PopFactor

'UPDATE osmEmissions

DoCmd.RunSQL "UPDATE osmEmissions INNER JOIN SumPopET2 ON osmEmissions.EquipmentTypeID = SumPopET2.EquipmentTypeID SET osmEmissions.BaseBSFC = [osmEmissions]![BaseBSFC][SumPopET2]![AdjustmentFraction], osmEmissions.BaseNOx = [osmEmissions]![BaseNOx]*[SumPopET2]![AdjustmentFraction], osmEmissions.BasePM = [osmEmissions]![BasePM]*[SumPopET2]![AdjustmentFraction], osmEmissions.BaseHC = [osmEmissions]![BaseHC]*[SumPopET2]![AdjustmentFraction], osmEmissions.ScenBSFC = [osmEmissions]![ScenBSFC]*[SumPopET2]![AdjustmentFraction], osmEmissions.ScenNOx = [osmEmissions]![ScenNOx]*[SumPopET2]![AdjustmentFraction], osmEmissions.ScenPM = [osmEmissions]![ScenPM]*[SumPopET2]![AdjustmentFraction], osmEmissions.ScenHC = [osmEmissions]![ScenHC]*[SumPopET2]![AdjustmentFraction];"*

''''''

'GrowEmissions - Growth factors were derived to reflect the growth industries will have into the future. These are available in the table 'GrowthFactors.' Growth factors are specific to calendar year and equipment class (sector).' Instead of attempting to grow populations or activities, growth is applied directly to emissions to reflect an overall growth.

'BSFC = BSFC * GrowthFactor, NOx = NOx * GrowthFactor, PM = PM * GrowthFactor, HC = HC * GrowthFactor,

'UPDATE osmEmissions

*DoCmd.RunSQL "UPDATE (GrowthFactor INNER JOIN osmEmissions ON GrowthFactor.CalendarYear = osmEmissions.CalendarYear) INNER JOIN LookupEquipmentType ON (LookupEquipmentType.[Equipment Class] = GrowthFactor.EquipmentClass) AND (osmEmissions.EquipmentTypeID = LookupEquipmentType.EquipmentTypeID) SET osmEmissions.BaseBSFC = osmEmissions!BaseBSFC*GrowthFactor!GrowthFactor, osmEmissions.BaseNOx = osmEmissions!BaseNOx*GrowthFactor!GrowthFactor, osmEmissions.BasePM = osmEmissions!BasePM*GrowthFactor!GrowthFactor, osmEmissions.BaseHC = osmEmissions!BaseHC*GrowthFactor!GrowthFactor, osmEmissions.ScenBSFC = osmEmissions!ScenBSFC*GrowthFactor!GrowthFactor, osmEmissions.ScenNOx = osmEmissions!ScenNOx*GrowthFactor!GrowthFactor, osmEmissions.ScenPM = osmEmissions!ScenPM*GrowthFactor!GrowthFactor, osmEmissions.ScenHC = osmEmissions!ScenHC*GrowthFactor!GrowthFactor;"*

'GramsToTons - Grams per year are converted to tons per year by factor = 1/(453.59237*2000)
'Update osmEmissions

*DoCmd.RunSQL "UPDATE osmEmissions SET osmEmissions.BaseNOx = [osmEmissions]![BaseNOx]/(453.59237*2000), osmEmissions.BasePM = [osmEmissions]![BasePM]/(453.59237*2000), osmEmissions.BaseHC = [osmEmissions]![BaseHC]/(453.59237*2000), osmEmissions.ScenNOx = [osmEmissions]![ScenNOx]/(453.59237*2000), osmEmissions.ScenPM = [osmEmissions]![ScenPM]/(453.59237*2000), osmEmissions.ScenHC = [osmEmissions]![ScenHC]/(453.59237*2000);"*

''''''''''''''''''''The table 'osmEmissions' is specific to all vehicles run through the program OSM. A spatial allocation of emissions is desired so the following 9 queries allocate emissions to air basins around California while at the same time formatting the final output to be used by the form 'UserForm.'

'BaselineAllocation1 - Emissions are summed without individual vehicle or fleet specificity: Grouped by Calendar Year, fleet size, equipment type, horsepower, and model year. Baseline is done separate from Scenario because results are model year specific and each has varying populations and emissions when grouped by model year. Results are put into the table 'osmEmBaseAll'.
'INTO osmEmBaseAll

DoCmd.RunSQL "SELECT osmEmissions.CalendarYear, osmEmissions.FleetSize, osmEmissions.EquipmentTypeID, osmEmissions.HorsepowerBin, osmEmissions.BaseModelYear, Sum(osmEmissions.BaseBSFC) AS SumOfBaseBSFC, Sum(osmEmissions.BaseNOx) AS SumOfBaseNOx, Sum(osmEmissions.BasePM) AS SumOfBasePM, Sum(osmEmissions.BaseHC) AS SumOfBaseHC INTO osmEmAllBase FROM osmEmissions GROUP BY osmEmissions.CalendarYear, osmEmissions.FleetSize, osmEmissions.EquipmentTypeID, osmEmissions.HorsepowerBin, osmEmissions.BaseModelYear;"

'ScenarioAllocation1 - Just as in BaselineAllocation1, emissions are summed by groups for location allocation. Results are put into a table 'osmEmScenAll'.
'INTO osmEmScenAll

DoCmd.RunSQL "SELECT osmEmissions.CalendarYear, osmEmissions.FleetSize, osmEmissions.EquipmentTypeID, osmEmissions.HorsepowerBin, osmEmissions.ScenModelYear, Sum(osmEmissions.ScenBSFC) AS SumOfScenBSFC, Sum(osmEmissions.ScenNOx) AS SumOfScenNOx, Sum(osmEmissions.ScenPM) AS SumOfScenPM, Sum(osmEmissions.ScenHC)

AS SumOfScenHC INTO osmEmAllScen FROM osmEmissions GROUP BY osmEmissions.CalendarYear, osmEmissions.FleetSize, osmEmissions.EquipmentTypeID, osmEmissions.HorsepowerBin, osmEmissions.ScenModelYear;"

'Now that baseline and scenario emissions have been separated, the following six queries combine the information back into one table for programming ease and saving memory. The first two queries count how many unique vehicle types (by calendar year, age, type, etc.) there are in the baseline table. These vehicle types don't exist in the scenario table and must be formatted to be included. The first query counts all vehicle types in baseline.

DoCmd.RunSQL "SELECT osmEmAllBase.CalendarYear, osmEmAllBase.FleetSize, osmEmAllBase.EquipmentTypeID, osmEmAllBase.HorsepowerBin, osmEmAllBase.BaseModelYear INTO osmEmAllBaseUnique FROM osmEmAllBase;"

'This second query deletes entries from baseline that also appear in scenario, thus leaving unique baseline entries.

DoCmd.RunSQL "DELETE DISTINCTROW osmEmAllBaseUnique. FROM osmEmAllBaseUnique INNER JOIN osmEmAllScen ON (osmEmAllBaseUnique.BaseModelYear = osmEmAllScen.ScenModelYear) AND (osmEmAllBaseUnique.HorsepowerBin = osmEmAllScen.HorsepowerBin) AND (osmEmAllBaseUnique.EquipmentTypeID = osmEmAllScen.EquipmentTypeID) AND (osmEmAllBaseUnique.FleetSize = osmEmAllScen.FleetSize) AND (osmEmAllBaseUnique.CalendarYear = osmEmAllScen.CalendarYear);"*

'The following query enters vehicle type information from the scenario table into a new table 'osmEmissionsForAirBasin1.' It also creates empty fields for scenario emissions information.

DoCmd.RunSQL "SELECT osmEmAllScen.CalendarYear, osmEmAllScen.FleetSize, osmEmAllScen.EquipmentTypeID, osmEmAllScen.HorsepowerBin, [osmEmAllScen]![ScenModelYear] AS ModelYear, CDbl(0) AS BaseBSFC, CDbl(0) AS BaseNOx, CDbl(0) AS BasePM, CDbl(0) AS BaseHC, CDbl(0) AS ScenBSFC, CDbl(0) AS ScenNOx, CDbl(0) AS ScenPM, CDbl(0) AS ScenHC INTO osmEmissionsForAirBasin1 FROM osmEmAllScen;"

'The following query enters the same vehicle information as above for the unique vehicle types in the baseline table.

DoCmd.RunSQL "INSERT INTO osmEmissionsForAirBasin1 (CalendarYear, FleetSize, EquipmentTypeID, HorsepowerBin, ModelYear, BaseBSFC, BaseNOx, BasePM, BaseHC, ScenBSFC, ScenNOx, ScenPM, ScenHC)

SELECT osmEmAllBaseUnique.CalendarYear, osmEmAllBaseUnique.FleetSize, osmEmAllBaseUnique.EquipmentTypeID, osmEmAllBaseUnique.HorsepowerBin, osmEmAllBaseUnique.BaseModelYear, 0 AS BaseBSFC, 0 AS BaseNOx, 0 AS BasePM, 0 AS BaseHC, 0 AS ScenBSFC, 0 AS ScenNOx, 0 AS ScenPM, 0 AS ScenHC FROM osmEmAllBaseUnique;"

'Baseline emissions are loaded.

DoCmd.RunSQL "UPDATE osmEmissionsForAirBasin1 INNER JOIN osmEmAllBase ON (osmEmissionsForAirBasin1.ModelYear = osmEmAllBase.BaseModelYear) AND (osmEmissionsForAirBasin1.HorsepowerBin = osmEmAllBase.HorsepowerBin) AND (osmEmissionsForAirBasin1.EquipmentTypeID = osmEmAllBase.EquipmentTypeID) AND (osmEmissionsForAirBasin1.FleetSize = osmEmAllBase.FleetSize) AND (osmEmissionsForAirBasin1.CalendarYear = osmEmAllBase.CalendarYear) SET osmEmissionsForAirBasin1.BaseBSFC = [osmEmAllBase]![SumOfBaseBSFC], osmEmissionsForAirBasin1.BaseNOx = [osmEmAllBase]![SumOfBaseNOx], osmEmissionsForAirBasin1.BasePM = [osmEmAllBase]![SumOfBasePM], osmEmissionsForAirBasin1.BaseHC = [osmEmAllBase]![SumOfBaseHC];"

'Scenario emissions are loaded

DoCmd.RunSQL "UPDATE osmEmissionsForAirBasin1 INNER JOIN osmEmAllScen ON (osmEmissionsForAirBasin1.ModelYear = osmEmAllScen.ScenModelYear) AND (osmEmAllScen.HorsepowerBin = osmEmissionsForAirBasin1.HorsepowerBin) AND (osmEmissionsForAirBasin1.EquipmentTypeID = osmEmAllScen.EquipmentTypeID) AND (osmEmAllScen.FleetSize = osmEmissionsForAirBasin1.FleetSize) AND (osmEmissionsForAirBasin1.CalendarYear = osmEmAllScen.CalendarYear) SET osmEmissionsForAirBasin1.ScenBSFC = [osmEmAllScen]![SumOfScenBSFC], osmEmissionsForAirBasin1.ScenNOx = [osmEmAllScen]![SumOfScenNOx], osmEmissionsForAirBasin1.ScenPM = [osmEmAllScen]![SumOfScenPM], osmEmissionsForAirBasin1.ScenHC = [osmEmAllScen]![SumOfScenHC];"

'This final query allocates both baseline and scenario emissions to air basins around California. Allocation factors are calendar year and equipment class (sector) specific. Each entry in osmEmissionsForAirBasin1 is allocated to the 15 different are basins.

DoCmd.RunSQL "SELECT osmEmissionsForAirBasin1.CalendarYear, osmEmissionsForAirBasin1.FleetSize, osmEmissionsForAirBasin1.EquipmentTypeID, osmEmissionsForAirBasin1.HorsepowerBin, osmEmissionsForAirBasin1.ModelYear, AirBasinAllocation.AirBasin,

```
osmEmissionsForAirBasin1!BaseBSFC*AirBasinAllocation!Allocation AS
BaseBSFC, osmEmissionsForAirBasin1!BaseNOx*AirBasinAllocation!Allocation
AS BaseNOx, osmEmissionsForAirBasin1!BasePM*AirBasinAllocation!Allocation
AS BasePM, osmEmissionsForAirBasin1!BaseHC*AirBasinAllocation!Allocation
AS BaseHC,
osmEmissionsForAirBasin1!ScenBSFC*AirBasinAllocation!Allocation AS
ScenBSFC, osmEmissionsForAirBasin1!ScenNOx*AirBasinAllocation!Allocation
AS ScenNOx, osmEmissionsForAirBasin1!ScenPM*AirBasinAllocation!Allocation
AS ScenPM, osmEmissionsForAirBasin1!ScenHC*AirBasinAllocation!Allocation
AS ScenHC INTO osmEmissionsForAirBasin2 " & _
   "FROM (osmEmissionsForAirBasin1 INNER JOIN LookupEquipmentType ON
osmEmissionsForAirBasin1.EquipmentTypeID =
LookupEquipmentType.EquipmentTypeID) INNER JOIN AirBasinAllocation ON
(osmEmissionsForAirBasin1.CalendarYear = AirBasinAllocation.CalendarYear)
AND (LookupEquipmentType.[Equipment Class] = AirBasinAllocation.[Equipment
Class]);"

''''''''

DoCmd.SetWarnings True

End Sub
```

Attachment B

Fleet Survey Results

The table below provides a summary of the results from the 1000 fleet survey used to determine the non-compliance rate for DOORS reporting. Staff used the results from this survey to determine the overall Statewide California off-road diesel population subject to the off-road diesel rule. Refer to section 1.C for a more detailed description of how these data were used to calculate the non-compliance estimate.

Survey Response	Number of Fleets
Agriculture (not subject to rule)	151
Already reported	97
Needs to report	73
No off-road vehicles	73
No Response	224
Could not be Reached	382
Total	1000

Attachment C

1.Air Resources Board

Linda S. Adams
Secretary for
Environmental Protection

Mary D. Nichols, Chairman
9480 Telstar Avenue, Suite 4
El Monte, California 91731 www.arb.ca.gov

TO: David Chou, Manager
Off-Road Modeling and Assessment Section

FROM: John Karim, Manager
In-Use Retrofit Section

DATE: April 1, 2010

SUBJECT: ESTIMATED LOAD FACTORS FOR THE SHOWCASE AND SUPPLEMENTAL ENVIRONMENTAL PROJECTS (SEP) EQUIPMENT

Background:

The In-Use Retrofit Section (IURS) has been involved in a number of projects, such as the Showcase and SEPs, to demonstrate the viability of diesel emission control devices in a variety of off-road environments as well as to evaluate new emission control systems that might be verified by the Air Resources Board (Board). In order to do so, IURS required the use of a robust and a vibration/shock resistant data-logger suitable for the off-road environment. The data-logger was specifically designed by staff to facilitate first hand data acquisition and storage of critical parameters for the evaluation of after-treatment devices. As a result of the data-logging of retrofitted equipment, an extensive data set is being compiled for the purpose of monitoring and verifying the operation of after-treatment devices for the off-road heavy-duty diesel fleet. This data set has been proven to be very valuable in providing real-world information about the off-road heavy-duty diesel fleet's load activities, annual operational hours, duty cycles, and other interesting trends that the Board would use for recommending regulatory efforts and updating inventory modeling factors. Staff were assigned the task of computing the load factors for the off-road fleet from the data set that was collected by IURS. This report includes the estimated load factors per fleet and the average estimated load factors per category.

Calculation Methodology:

The data-logger collects real-time engine and exhaust data to ensure the operation of the after-treatment device. The data set includes exhaust temperature and engine speed, and additional engine parameters when the equipment has an Electronic Control Unit (ECU). The data set was reviewed and screened for accuracy and the estimated load factors were calculated. The exhaust temperature and engine speed data were used as surrogates for estimating percentage load per instant. In addition, information from equipment that reported ECU load data were analyzed and average load data was calculated. The following equation was applied to estimate the percent load value:

Estimated Percent Load per second = $(S - S_{min} / S_{max} - S_{min}) * 100$
where S is the surrogate data (exhaust temperature or engine speed).

Data Set:

The evaluated data sets consisted of 60 retrofitted and data-logged pieces of equipment. Load factor estimation was computed for all 60 based on exhaust temperature. Out of these 60, 52 also had engine speed data for estimation, and 10 also had ECU percent load data used for analysis. These data sets were used to calculate the estimated load factors. The data analysis consisted of detailed review and calculation of about 630 data files. The table below shows the categorical breakdown of the data set:

No.	Category	Year Range	Year Average	Year Mode	HP Range	HP Average	Tier Range	Tier Mode	No. of Vehicles	No. of Downloads
1	Wheel Loader	2004	2004	2004	272	272	2	2	1	5
2	Tractor-Loader-Backhoe	1996-2006	2000	1999	47-362	135.1	0-2	1	9	65
3	Scraper	2000-2001	2000	2000	457-632	545.6	1-2	1	7	77
4	Rubber Tired Dozer	1982	1982	1982	255	255	0	0	1	1
5	Rubber Tired Loader	1992-2004	1997	1995	95-375	214	0-2	0	11	93
6	Rough Terrain Forklift	1999-2005	2001	N/A	107-200	147	1-2	1	4	38
7	Other Mobile Off-Road	2000-2006	2003	2000	77-439	146.8	1-2	1	6	19
8	Off-Highway Tractor	2006-2007	2007	2007	125-140	130	2-3	3	9	167
9	Grader	2000-2004	2003	2004	185-225	197.3	1-2	2	4	29
10	Excavator	1994-2002	1999	2002	148-439	241.7	0-2	1	6	115
11	Crawler Tractor	1998-2001	2000	N/A	255-370	312.5	1	1	2	24

Summary:

After review and evaluation of the IURS data set, it was concluded that in general the load factors based on the two surrogates produced comparable results to those in the 2007 Off-Road Model; however, the ECU data appear to yield lower factors than this Model. There were two equipment categories, the Rough Terrain Forklift and the Grader, where the Model was 62% to 67% higher than what was computed using the surrogates. Since the estimates in the Model were 69% to 192% higher, respectively, than the average load factors computed using the ECU data in the two instances where ECU data were available, staff evaluated additional data from the Heavy-duty Diesel Laboratory in Downtown Los Angeles and from field tests that were conducted by IURS using their Portable Emission Measurement System (PEMS). The analysis of these additional data (not shown in this report) confirmed the observed trend showing the ECU data with lower load factors than the surrogates. The reliability of the ECU data was also confirmed by the Measurement Allowance Program, which concluded that load information reported by the ECU was reliable and may be used for the In-Use Compliance Verification Program. The table below summarizes the load factor data:

Category	2007 Off-Road Model Load Factor	IURS Load Factor			COMPARISON (% difference)		
		Temperature Based	RPM Based	ECU Based	2007 Off-Road Model vs. Temperature Based	2007 Off-Road Model vs. RPM Based	2007 Off-Road Model vs. ECU
Wheel Loader Avg. HP 272	55	60	45	N/A	-9%	+22%	N/A
Tractor-Loader-Backhoe Avg. HP 135	55	42	45	N/A	+32%	+21%	N/A
Scraper Avg. HP 546	72	56	58	N/A	+28%	+24%	N/A
Rubber Tired Dozer Avg. HP 255	59	59	57	N/A	+1%	+3%	N/A
Rubber Tired Loader Avg. HP 214	54	52	46	N/A	+5%	+16%	N/A
Rough Terrain Forklift Avg. HP 147	60	37	36	21	+63%	+67%	+192%
Other Mobile Off-Road Avg. HP 147	62	49	67	N/A	+27%	-7%	N/A
Off-Highway Tractor Avg. HP 130	65	56	59	38	+16%	+10%	+69%
Grader Avg. HP 197	61	38	47	N/A	+62%	+30%	N/A
Excavator Avg. HP 242	57	61	64	N/A	-7%	-11%	N/A
Crawler Tractor Avg. HP 313	64	70	61	N/A	-9%	+5%	N/A

Twelve figures are attached to this memorandum to show these findings. Figure 1 shows the IURS load factors and the 2007 Off-Road Model's load factors for all equipment categories. The subsequent figures (Fig. 2 to 12) show equipment load activity per category. The data set is available upon request.

The ever growing IURS data set, as a result of additional equipment being retrofitted under the Showcase and the SEP programs, provides a true snapshot of the off-road heavy-duty diesel fleet activities in California for both the private and the public sectors. Staff is working with equipment manufacturers to find solutions for acquiring ECU data from non-standard CAN Bus communication protocols to improve the IURS data set. Continued evaluation of these data would assist the Board in its efforts to reduce harmful emissions for this sector.

In conclusion, utilizing estimated load factors using surrogates could be overestimating the actual engine load by 69% to 192%, respectively, for the off-road fleet. Should you need additional information, please contact me at (626) 459-4303 or via email at jkarim@arb.ca.gov.

.

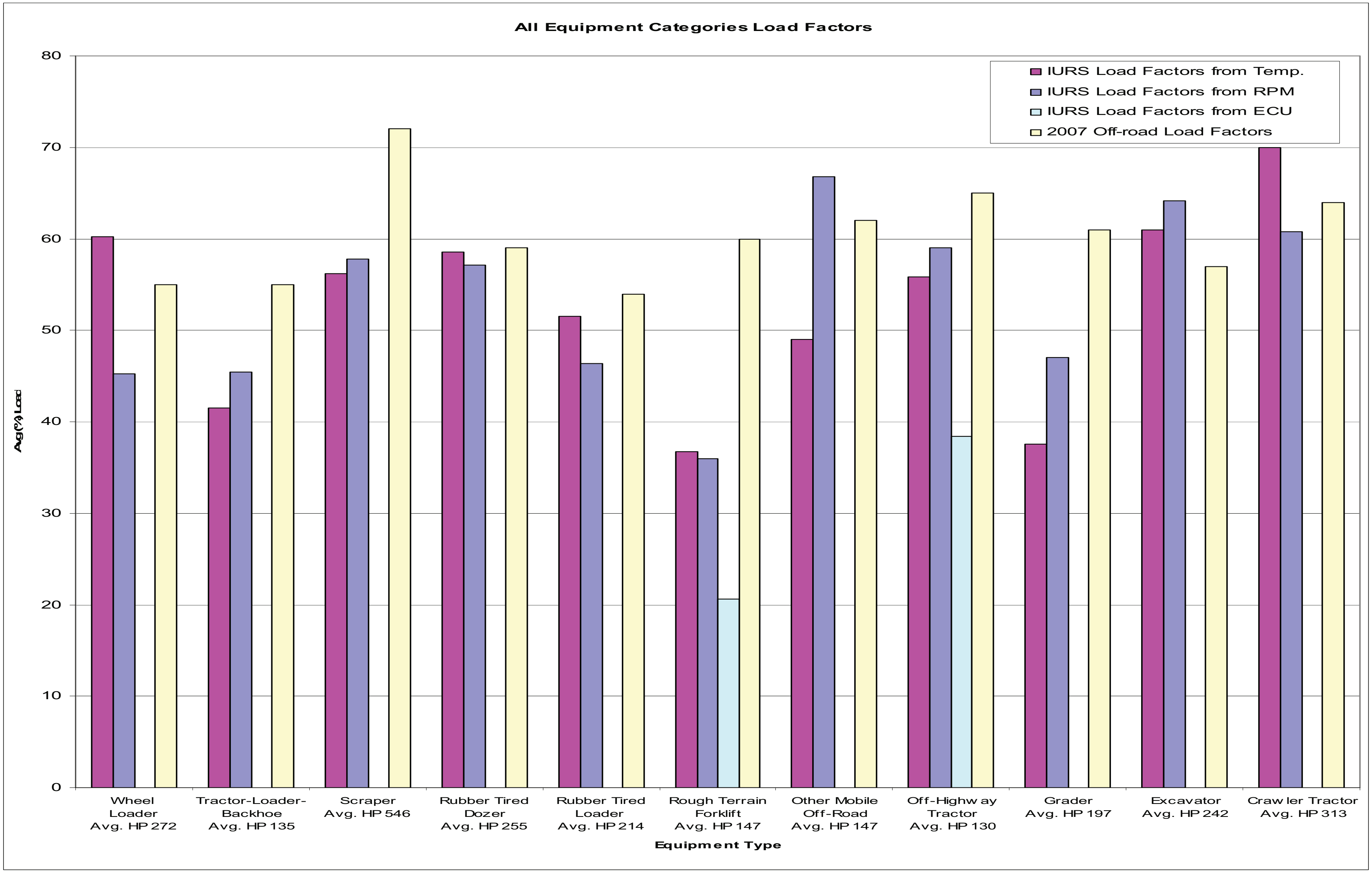


Figure 1. Load Factors for All Categories

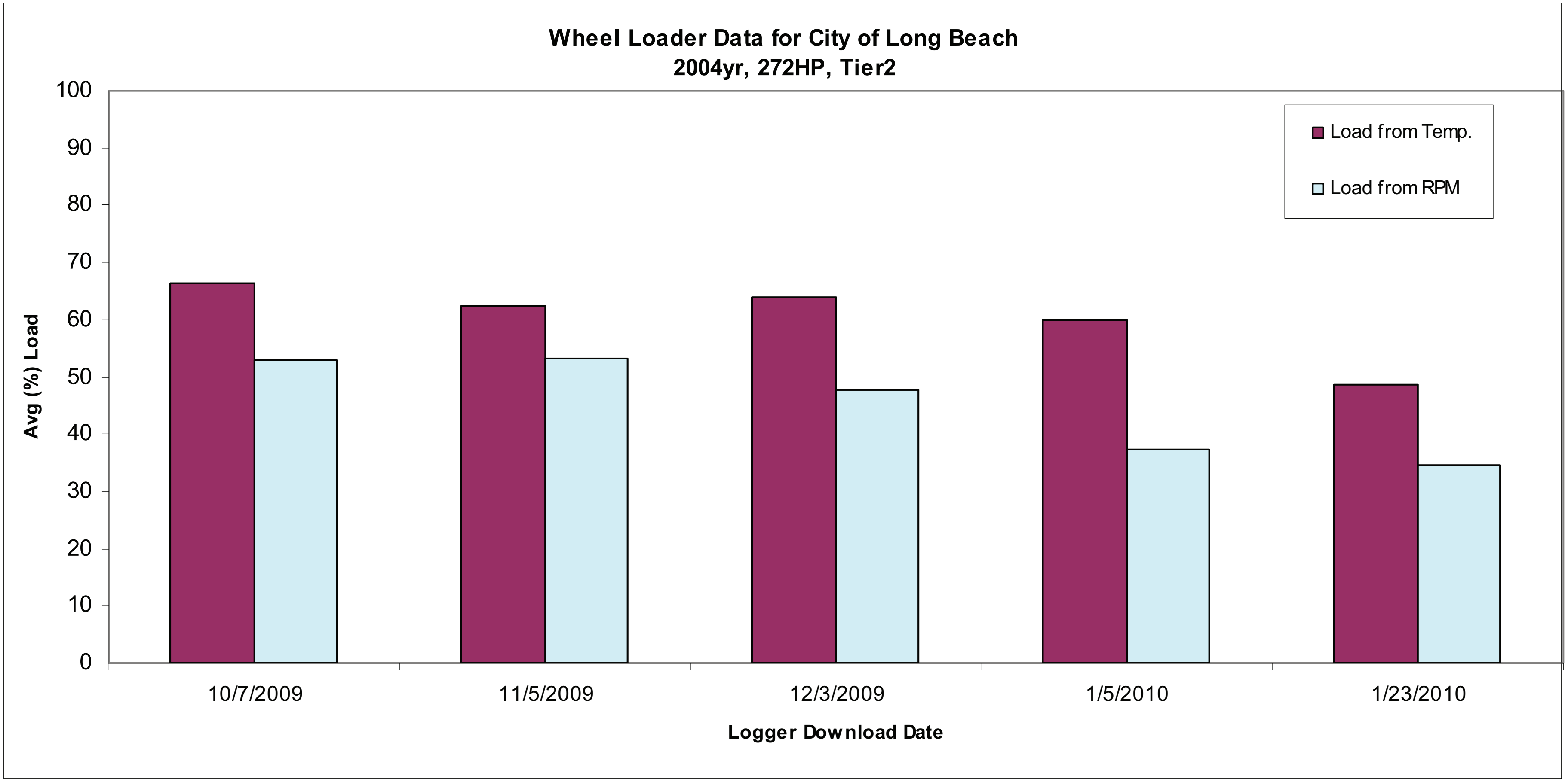


Figure 2. Wheel Loader Load Activity per Fleet

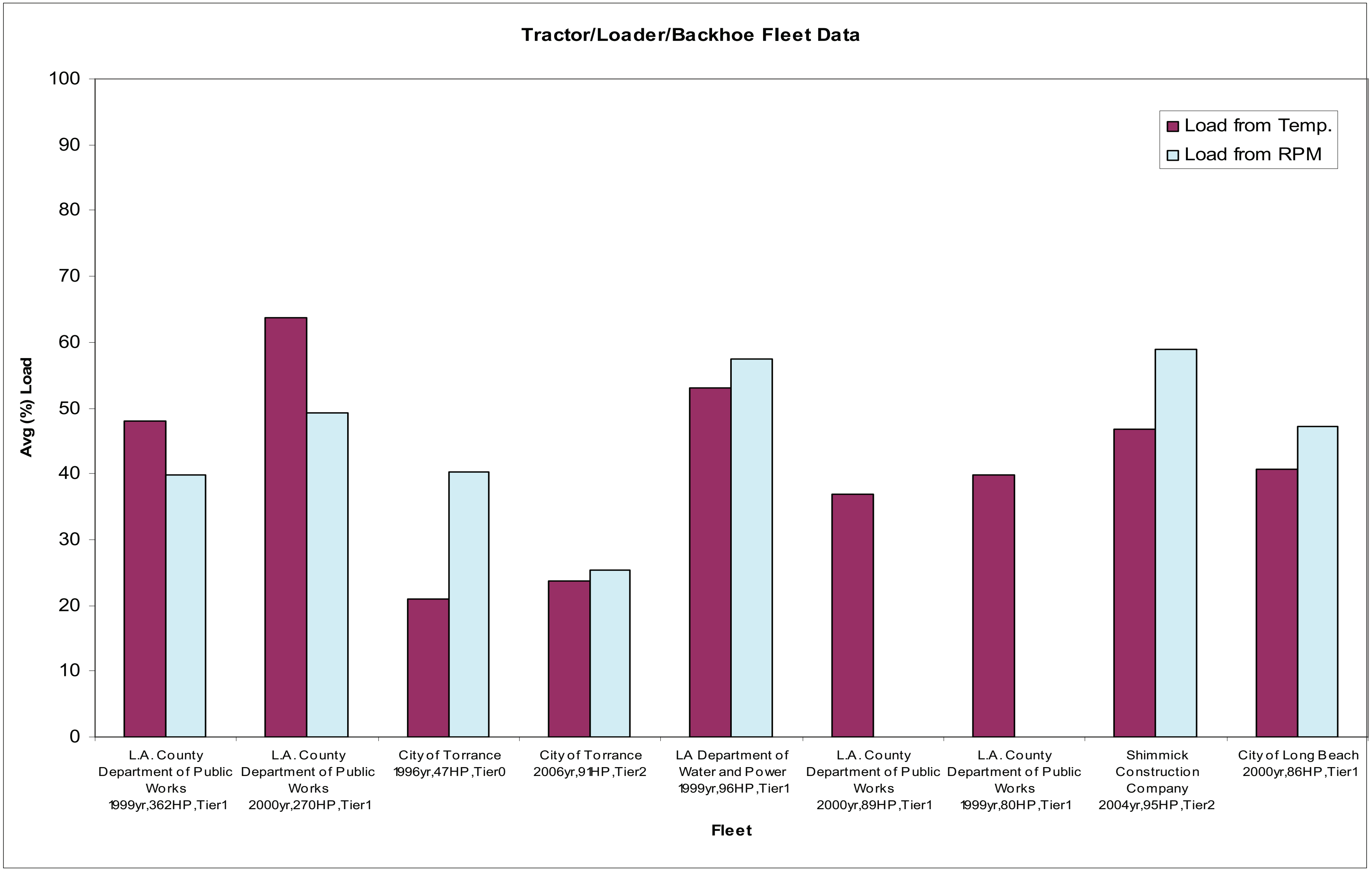


Figure 3. Tractor/Loader/Backhoe Load Activity per Fleet

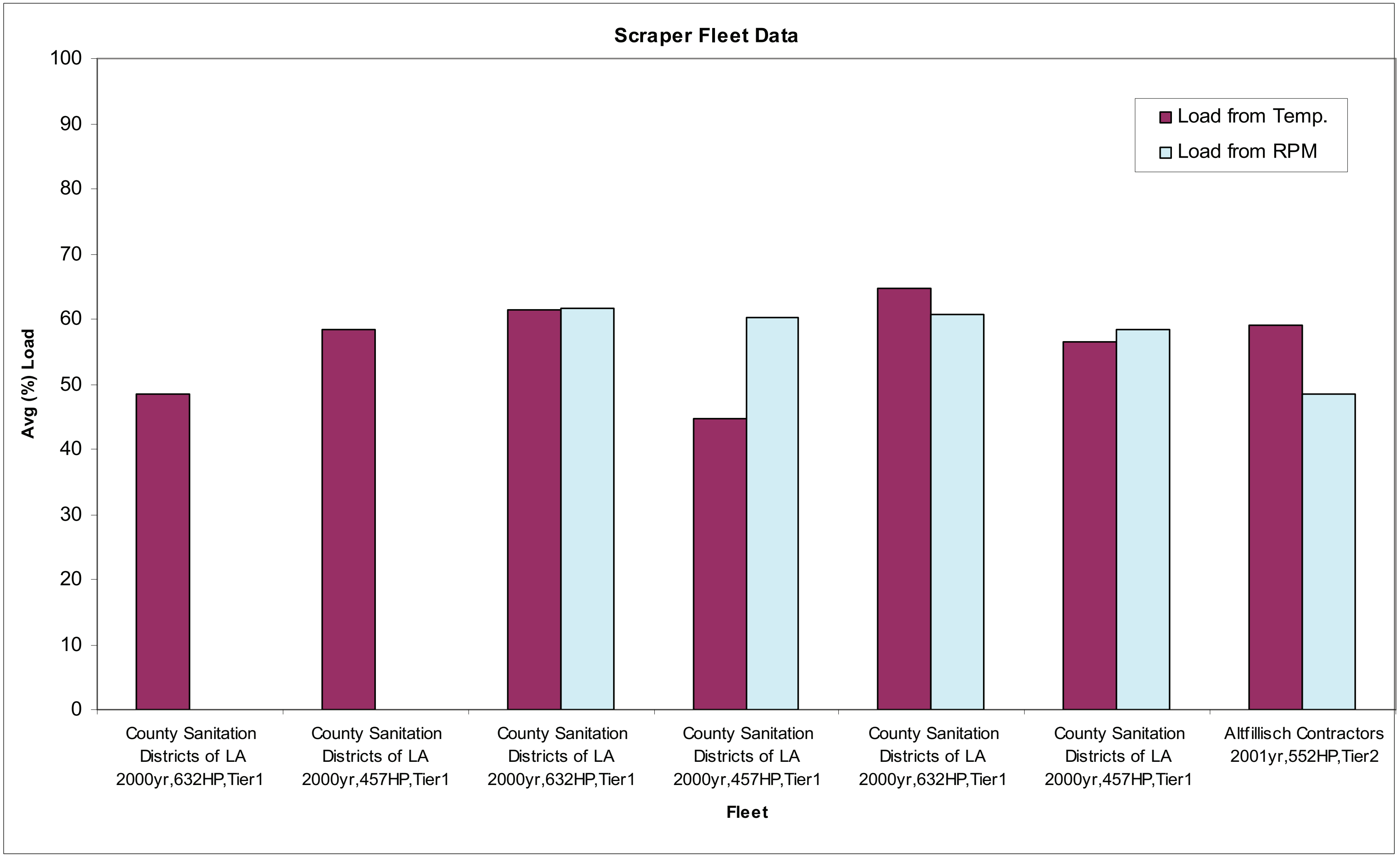


Figure 4. Scraper Load Activity per Fleet

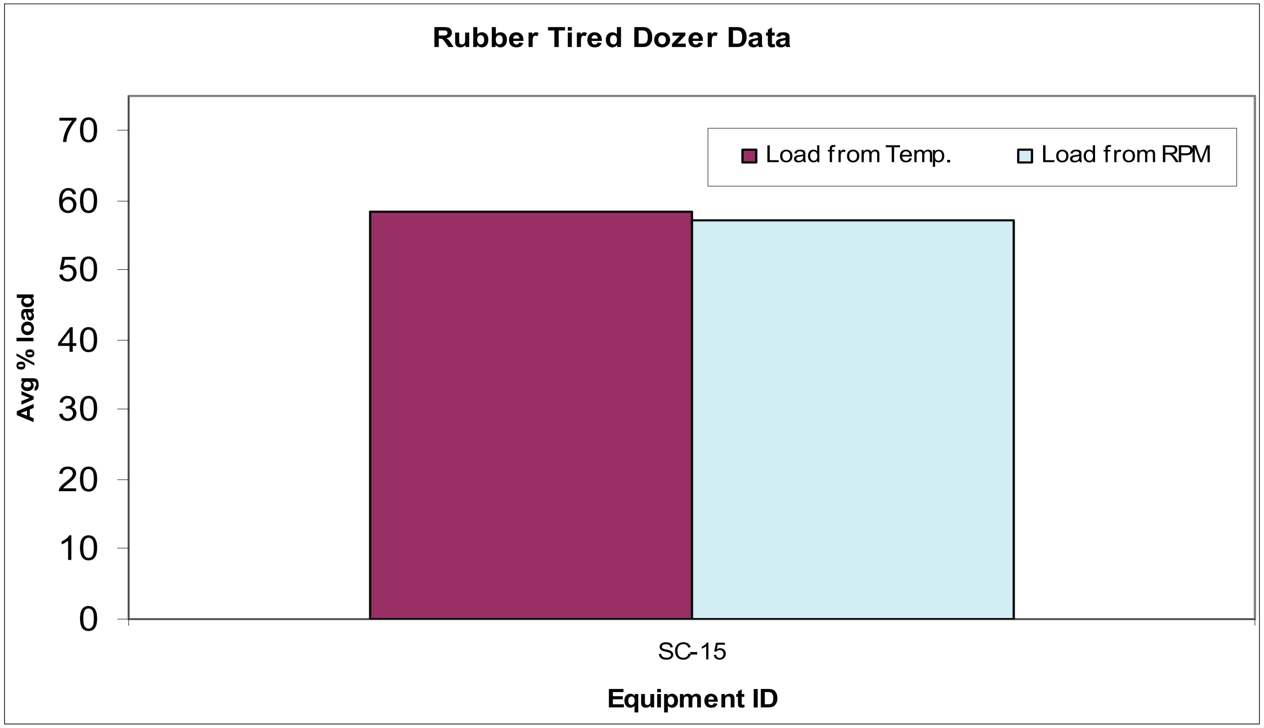


Figure 5. Rubber Tired Dozer Load Activity

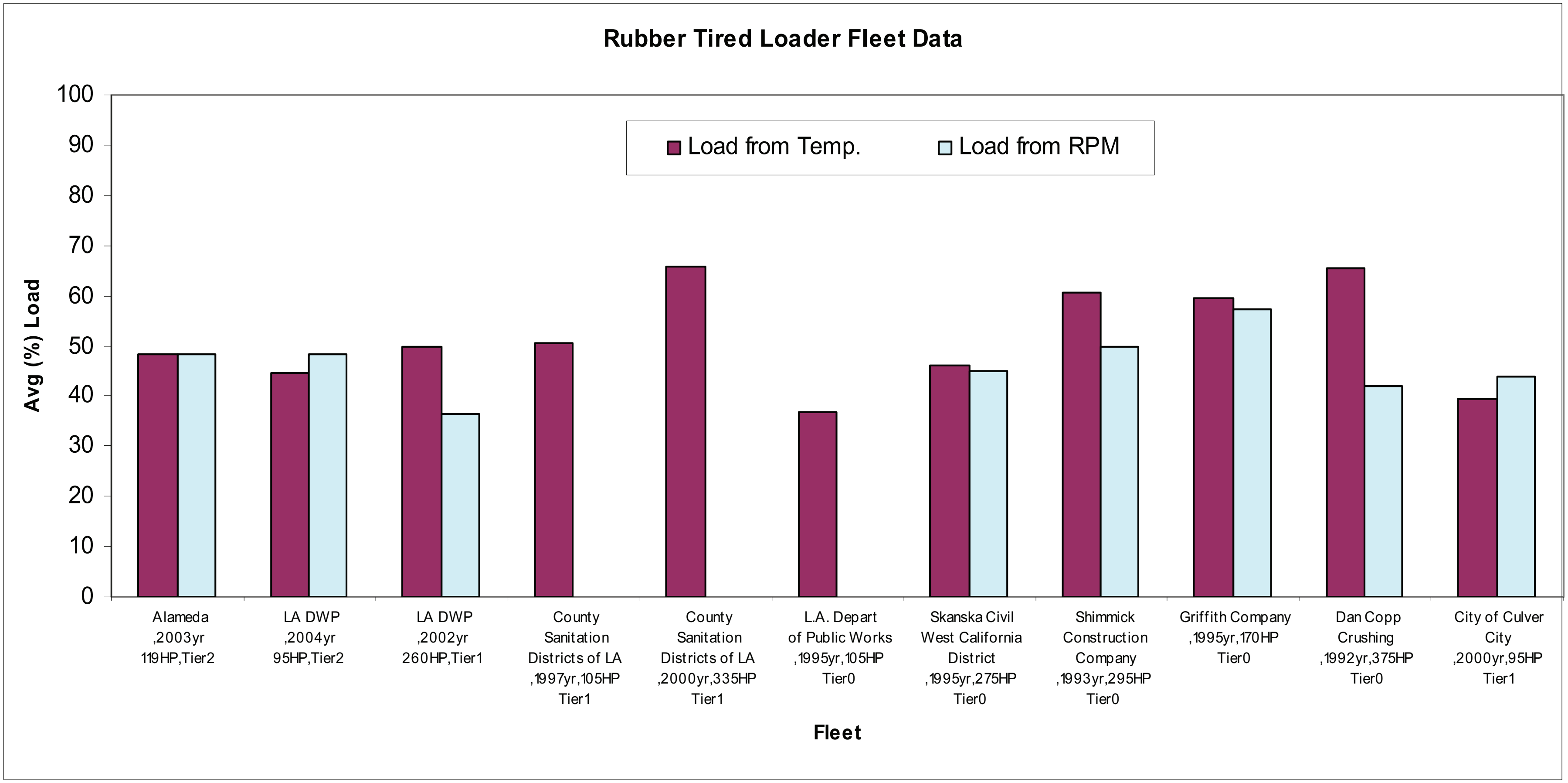


Figure 6. Rubber Tired Loader Load Activity per Fleet

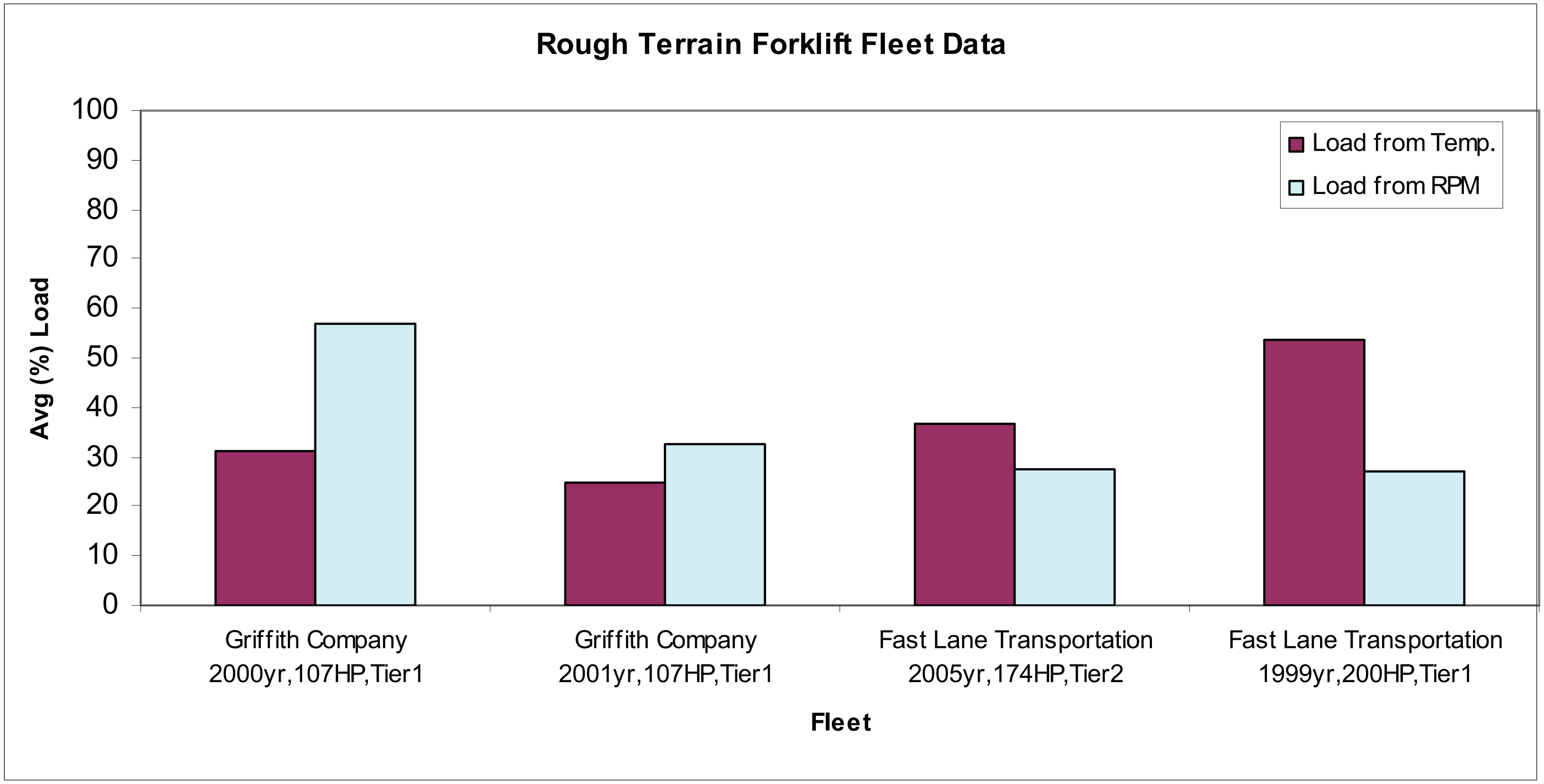


Figure 7. Rough Terrain Forklift Load Activity per Fleet

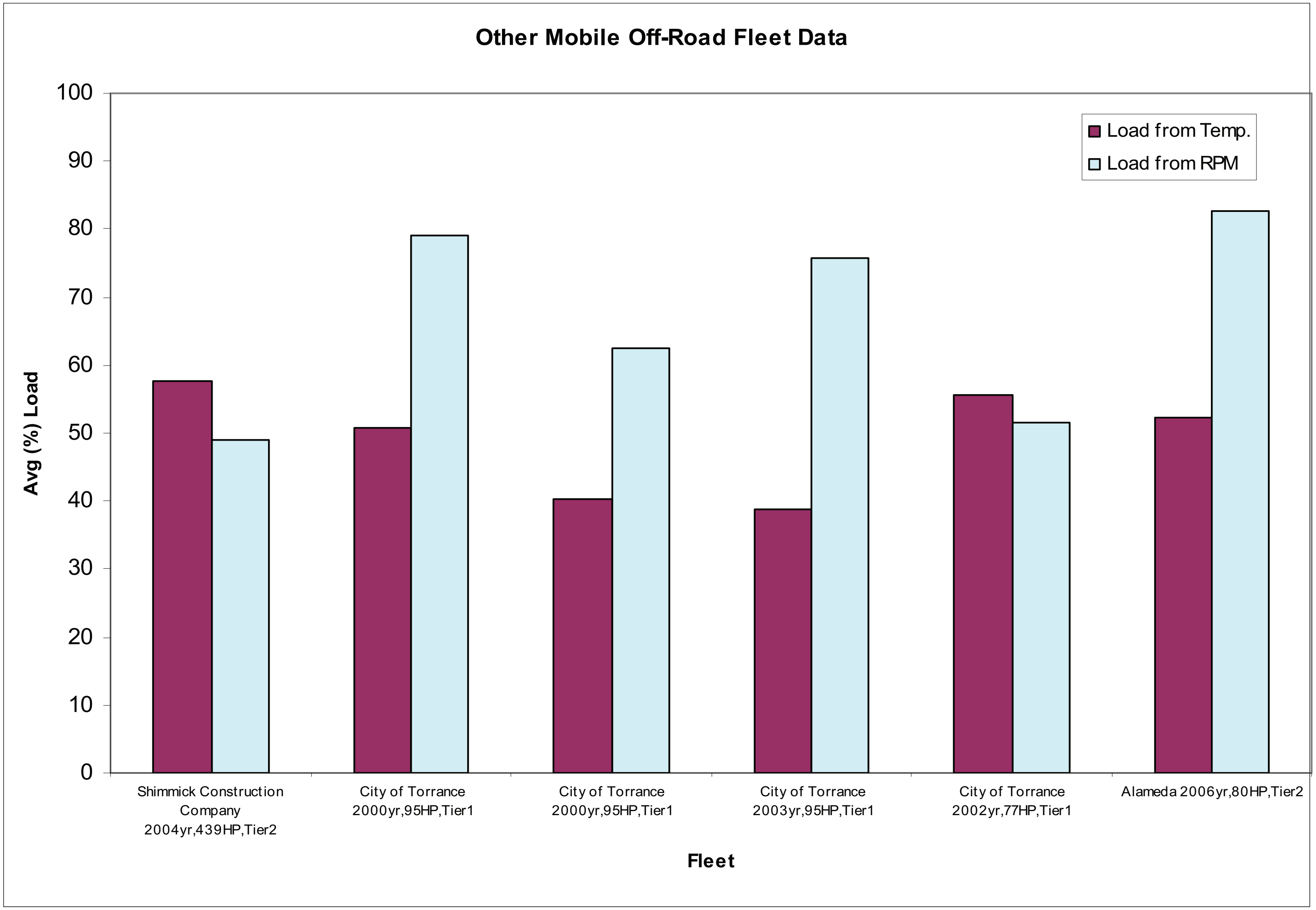


Figure 8. Other Mobile Off-Road Load Activity per Fleet

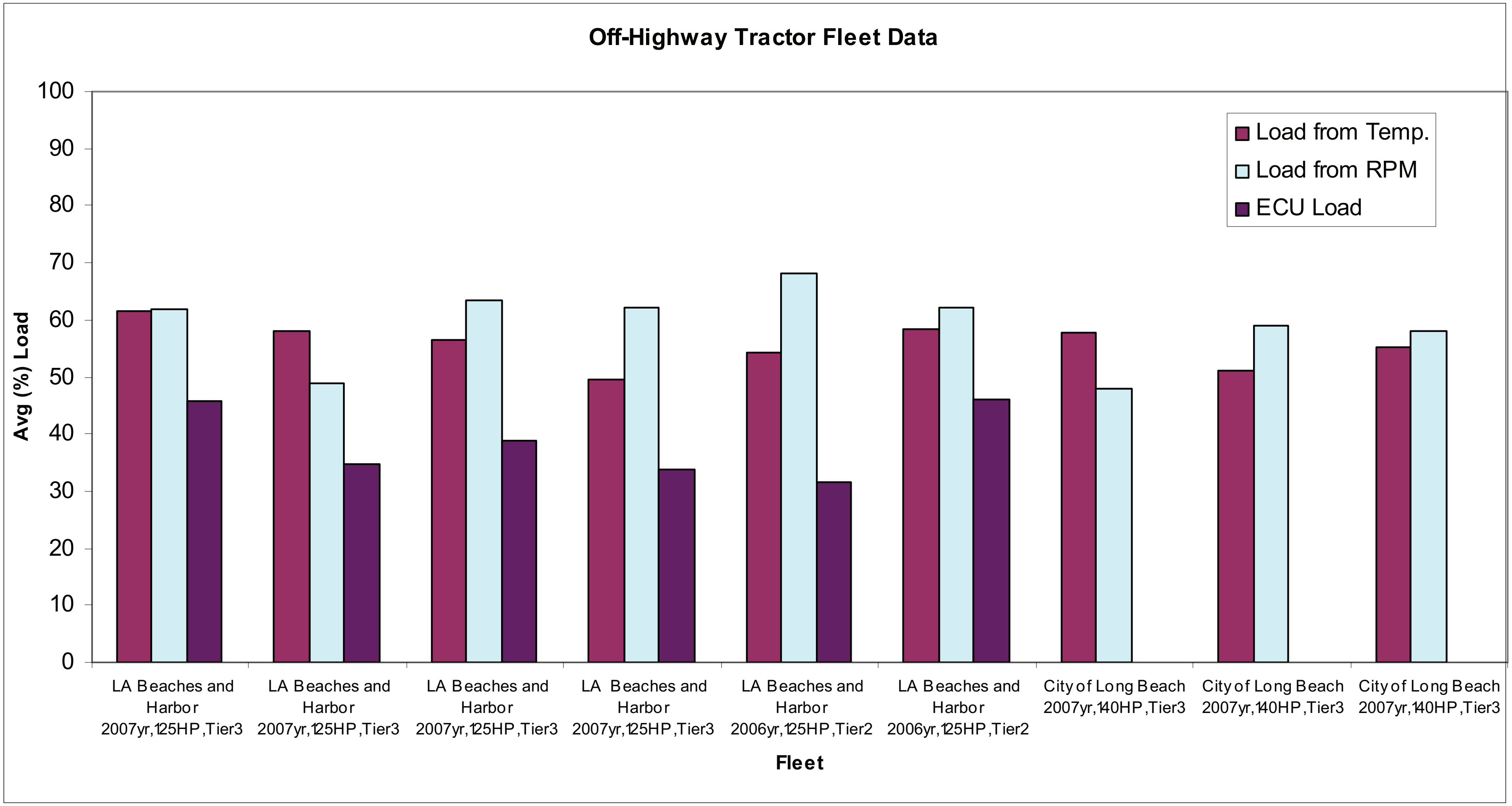


Figure 9. Off-Highway Tractor Load Activity per Fleet

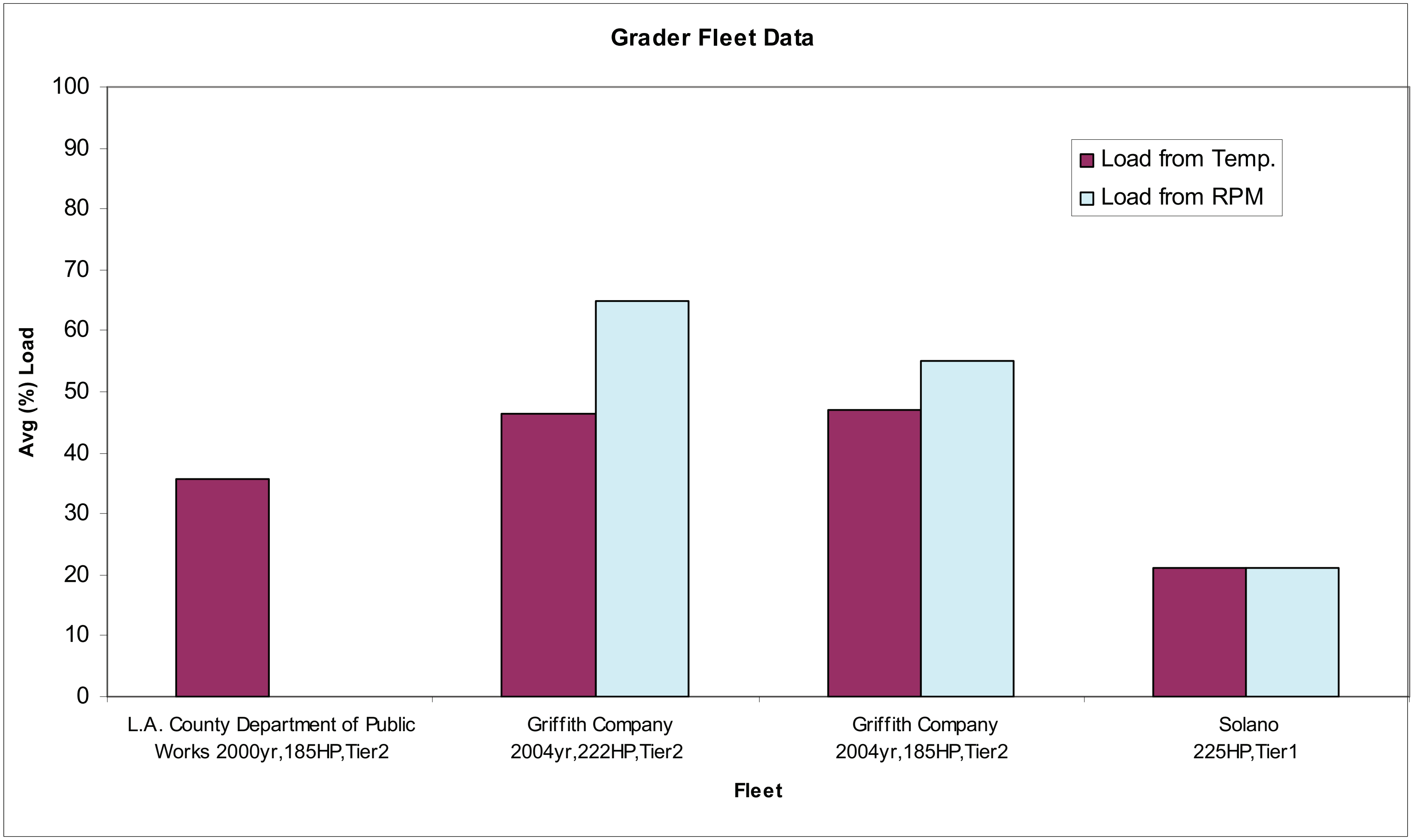


Figure 10. Grader Load Activity per Fleet

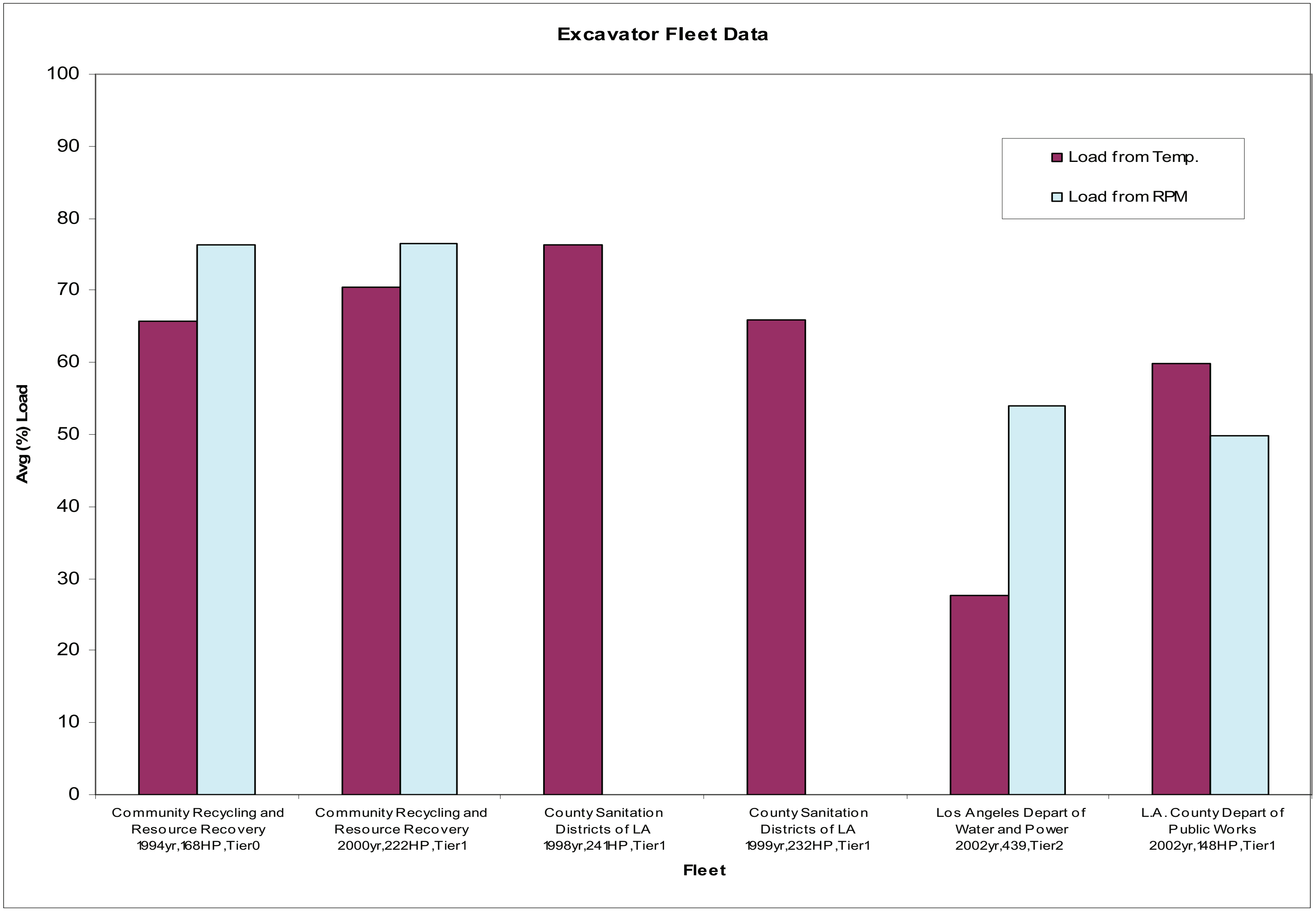


Figure 11. Excavator Load Activity per Fleet

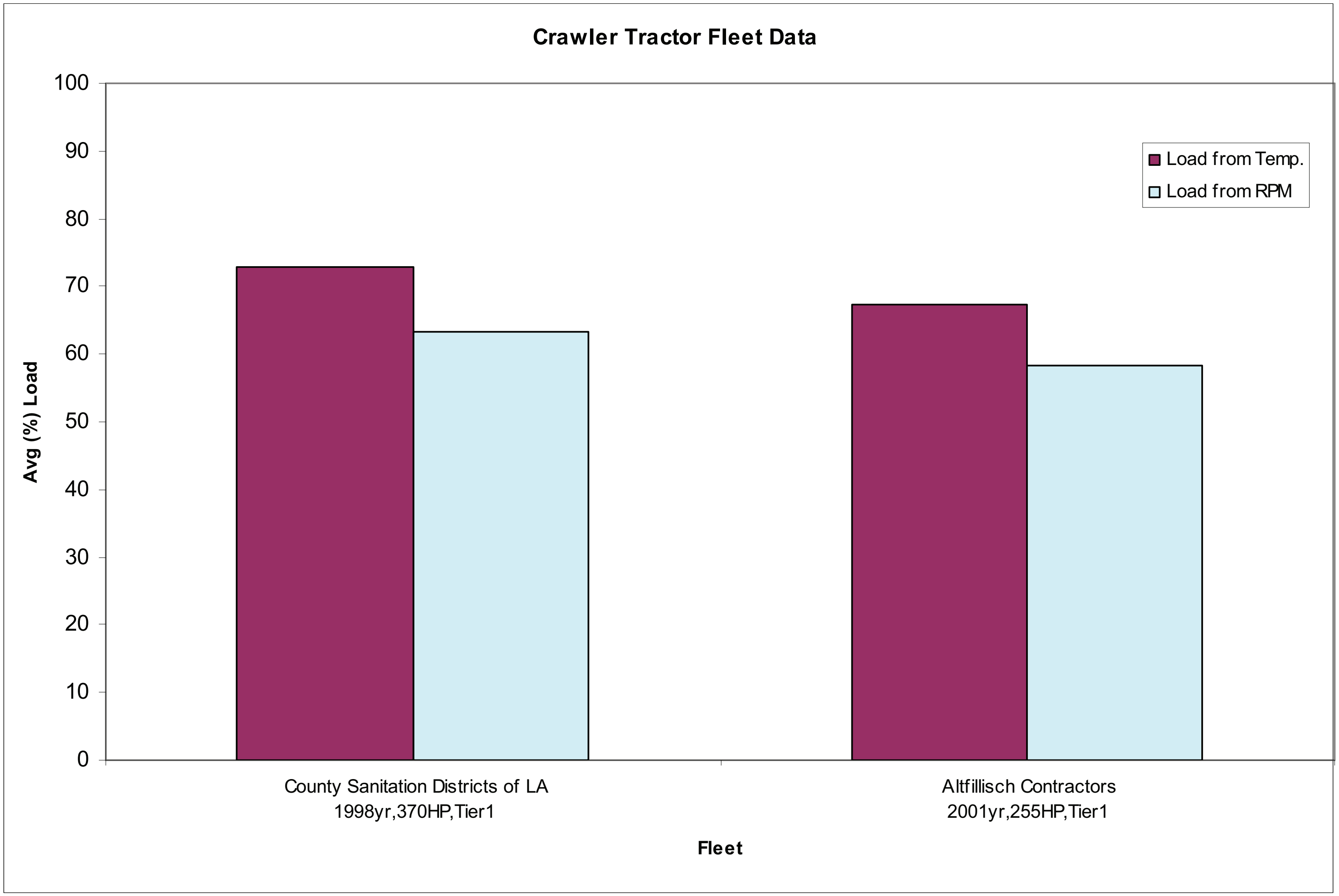


Figure 12. Crawler Tractor Load Activity per Fleet

Attachment D

Diesel Emission Factors (g/bhp-hr)

HP	Year	HC ZH (g/bhp-hr)	HC DR (g/bhp-hr^2)	CO ZH (g/bhp-hr)	CO DR (g/bhp-hr^2)	NOX ZH (g/bhp-hr)	NOX DR (g/bhp-hr^2)	PM ZH (g/bhp-hr)	PM DR (g/bhp-hr^2)
50	1987	1.84	2.35E-04	5.00	5.13E-04	7.00	1.05E-04	0.76	5.89E-05
50	1998	1.80	2.30E-04	5.00	5.13E-04	6.90	1.04E-04	0.76	5.89E-05
50	2003	1.45	1.85E-04	4.10	4.20E-04	5.55	1.03E-04	0.60	4.65E-05
50	2004	0.64	9.80E-05	3.27	3.34E-04	5.10	9.33E-05	0.43	3.36E-05
50	2005	0.37	6.90E-05	3.00	3.05E-04	4.95	9.67E-05	0.38	2.93E-05
50	2007	0.24	5.45E-05	2.86	2.90E-04	4.88	9.83E-05	0.35	2.72E-05
50	2012	0.10	4.00E-05	2.72	2.76E-04	4.80	1.00E-04	0.16	1.22E-05
50	2040	0.10	4.00E-05	2.72	2.76E-04	2.90	6.04E-05	0.01	1.11E-06
120	1987	1.44	6.66E-05	4.80	1.27E-04	13.00	3.01E-04	0.84	6.11E-05
120	1997	0.99	4.58E-05	3.49	9.23E-05	8.75	2.02E-04	0.69	5.02E-05
120	2003	0.99	4.58E-05	3.49	9.23E-05	6.90	1.60E-04	0.69	5.02E-05
120	2004	0.46	3.33E-05	3.23	8.55E-05	5.64	1.03E-04	0.39	2.85E-05
120	2005	0.28	2.92E-05	3.14	8.33E-05	5.22	8.40E-05	0.29	2.12E-05
120	2007	0.19	2.71E-05	3.09	8.21E-05	5.01	7.45E-05	0.24	1.76E-05
120	2011	0.10	2.50E-05	3.05	8.10E-05	2.89	3.80E-05	0.20	1.45E-05
120	2012	0.09	2.31E-05	3.05	8.10E-05	2.53	3.33E-05	0.07	4.69E-06
120	2014	0.09	2.31E-05	3.05	8.10E-05	2.53	3.33E-05	0.01	9.33E-07
120	2040	0.07	1.74E-05	3.05	8.10E-05	1.40	1.84E-05	0.01	9.33E-07
175	1969	1.32	6.11E-05	4.40	1.16E-04	14.00	3.24E-04	0.77	5.60E-05
175	1971	1.10	5.09E-05	4.40	1.16E-04	13.00	3.01E-04	0.66	4.80E-05

175	1979	1.00	4.63E-05	4.40	1.16E-04	12.00	2.78E-04	0.55	4.00E-05
175	1984	0.94	4.35E-05	4.30	1.14E-04	11.00	2.54E-04	0.55	4.00E-05
175	1987	0.88	4.07E-05	4.20	1.11E-04	11.00	2.54E-04	0.55	4.00E-05
175	1996	0.68	3.15E-05	2.70	7.14E-05	8.17	1.89E-04	0.38	2.76E-05
175	2002	0.68	3.15E-05	2.70	7.14E-05	6.90	1.60E-04	0.38	2.76E-05
175	2003	0.33	2.79E-05	2.70	7.14E-05	5.26	9.64E-05	0.24	1.70E-05
175	2004	0.22	2.63E-05	2.70	7.14E-05	4.72	7.52E-05	0.19	1.35E-05
175	2006	0.16	2.57E-05	2.70	7.14E-05	4.44	6.46E-05	0.16	1.18E-05
175	2011	0.10	2.50E-05	2.70	7.14E-05	2.45	3.20E-05	0.14	1.00E-05
175	2014	0.09	2.17E-05	2.70	7.14E-05	2.27	2.96E-05	0.01	4.67E-07
175	2040	0.05	1.17E-05	2.70	7.14E-05	0.27	3.56E-06	0.01	4.67E-07
250	1969	1.32	6.11E-05	4.40	1.16E-04	14.00	3.24E-04	0.77	5.60E-05
250	1971	1.10	5.09E-05	4.40	1.16E-04	13.00	3.01E-04	0.66	4.80E-05
250	1979	1.00	4.63E-05	4.40	1.16E-04	12.00	2.78E-04	0.55	4.00E-05
250	1984	0.94	4.35E-05	4.30	1.14E-04	11.00	2.54E-04	0.55	4.00E-05
250	1987	0.88	4.07E-05	4.20	1.11E-04	11.00	2.54E-04	0.55	4.00E-05
250	1995	0.68	3.15E-05	2.70	7.14E-05	8.17	1.89E-04	0.38	2.76E-05
250	2002	0.32	1.48E-05	0.92	2.43E-05	6.25	1.45E-04	0.15	7.96E-06
250	2003	0.19	2.09E-05	0.92	2.43E-05	5.00	9.05E-05	0.12	6.51E-06
250	2004	0.14	2.30E-05	0.92	2.43E-05	4.58	7.23E-05	0.11	6.03E-06
250	2006	0.12	2.40E-05	0.92	2.43E-05	4.38	6.33E-05	0.11	5.79E-06
250	2010	0.10	2.50E-05	0.92	2.43E-05	2.45	3.18E-05	0.11	5.59E-06
250	2013	0.07	1.83E-05	0.92	2.43E-05	1.36	1.77E-05	0.01	4.55E-07

250	2040	0.05	1.17E-05	0.92	2.43E-05	0.27	3.56E-06	0.01	4.55E-07
500	1969	1.26	4.39E-05	4.20	8.32E-04	14.00	2.33E-04	0.74	3.93E-05
500	1971	1.05	3.66E-05	4.20	8.32E-04	13.00	2.16E-04	0.63	3.34E-05
500	1979	0.95	3.31E-05	4.20	8.32E-04	12.00	2.00E-04	0.53	2.81E-05
500	1984	0.90	3.14E-05	4.20	8.32E-04	11.00	1.83E-04	0.53	2.81E-05
500	1987	0.84	2.93E-05	4.10	8.12E-04	11.00	1.83E-04	0.53	2.81E-05
500	1995	0.68	2.37E-05	2.70	5.35E-05	8.17	1.36E-04	0.38	2.02E-05
500	2000	0.32	1.12E-05	0.92	1.82E-05	6.25	1.04E-04	0.15	7.96E-06
500	2001	0.19	1.95E-05	0.92	1.82E-05	4.95	7.34E-05	0.12	6.51E-06
500	2002	0.14	2.22E-05	0.92	1.82E-05	4.51	6.32E-05	0.11	6.03E-06
500	2004	0.12	2.36E-05	0.92	1.82E-05	4.29	5.81E-05	0.11	5.79E-06
500	2005	0.10	2.50E-05	0.92	1.82E-05	4.00	5.30E-05	0.11	5.55E-06
500	2010	0.10	2.50E-05	0.92	1.82E-05	2.45	3.18E-05	0.11	5.55E-06
500	2013	0.07	1.83E-05	0.92	1.82E-05	1.36	1.77E-05	0.01	4.55E-07
500	2040	0.05	1.17E-05	0.92	1.82E-05	0.27	3.56E-06	0.01	4.55E-07
750	1969	1.26	4.39E-05	4.20	8.32E-04	14.00	2.33E-04	0.74	3.93E-05
750	1971	1.05	3.66E-05	4.20	8.32E-04	13.00	2.16E-04	0.63	3.34E-05
750	1979	0.95	3.31E-05	4.20	8.32E-04	12.00	2.00E-04	0.53	2.81E-05
750	1984	0.90	3.14E-05	4.20	8.32E-04	11.00	1.83E-04	0.53	2.81E-05
750	1987	0.84	2.93E-05	4.10	8.12E-04	11.00	1.83E-04	0.53	2.81E-05
750	1995	0.68	2.37E-05	2.70	5.35E-05	8.17	1.36E-04	0.38	2.02E-05
750	2001	0.32	1.12E-05	0.92	1.82E-05	6.25	1.04E-04	0.15	7.96E-06
750	2002	0.19	1.95E-05	0.92	1.82E-05	4.95	7.34E-05	0.12	6.51E-06

750	2003	0.14	2.22E-05	0.92	1.82E-05	4.51	6.32E-05	0.11	6.03E-06
750	2005	0.12	2.36E-05	0.92	1.82E-05	4.29	5.81E-05	0.11	5.79E-06
750	2010	0.10	2.50E-05	0.92	1.82E-05	2.45	3.18E-05	0.11	5.55E-06
750	2013	0.07	1.83E-05	0.92	1.82E-05	1.36	1.77E-05	0.01	4.55E-07
750	2040	0.05	1.17E-05	0.92	1.82E-05	0.27	3.56E-06	0.01	4.55E-07
1000	1969	1.26	4.39E-05	4.20	8.32E-04	14.00	2.33E-04	0.74	3.93E-05
1000	1971	1.05	3.66E-05	4.20	8.32E-04	13.00	2.16E-04	0.63	3.34E-05
1000	1979	0.95	3.31E-05	4.20	8.32E-04	12.00	2.00E-04	0.53	2.81E-05
1000	1984	0.90	3.14E-05	4.20	8.32E-04	11.00	1.83E-04	0.53	2.81E-05
1000	1987	0.84	2.93E-05	4.10	8.12E-04	11.00	1.83E-04	0.53	2.81E-05
1000	1999	0.68	1.12E-05	2.70	5.35E-05	8.17	1.36E-04	0.38	2.02E-06
1000	2005	0.32	1.12E-05	0.92	1.82E-05	6.25	1.04E-04	0.15	7.96E-06
1000	2006	0.19	1.95E-05	0.92	1.82E-05	4.95	7.34E-05	0.12	6.51E-06
1000	2007	0.14	2.22E-05	0.92	1.82E-05	4.51	6.32E-05	0.11	6.03E-06
1000	2009	0.12	2.36E-05	0.92	1.82E-05	4.29	5.81E-05	0.11	5.79E-06
1000	2010	0.10	2.50E-05	0.92	1.82E-05	4.08	5.30E-05	0.11	5.55E-06
1000	2014	0.07	1.83E-05	0.92	1.82E-05	2.36	3.06E-05	0.06	2.78E-06
1000	2040	0.05	1.17E-05	0.92	1.82E-05	2.36	3.06E-05	0.02	1.11E-06
9999	1969	1.26	4.39E-05	4.20	8.32E-04	14.00	2.33E-04	0.74	3.93E-05
9999	1971	1.05	3.66E-05	4.20	8.32E-04	13.00	2.16E-04	0.63	3.34E-05
9999	1979	0.95	3.31E-05	4.20	8.32E-04	12.00	2.00E-04	0.53	2.81E-05
9999	1984	0.90	3.14E-05	4.20	8.32E-04	11.00	1.83E-04	0.53	2.81E-05
9999	1987	0.84	2.93E-05	4.10	8.12E-04	11.00	1.83E-04	0.53	2.81E-05

9999	1999	0.68	1.12E-05	2.70	5.35E-05	8.17	1.36E-04	0.38	2.02E-06
9999	2005	0.32	1.12E-05	0.92	1.82E-05	6.25	1.04E-04	0.15	7.96E-06
9999	2006	0.19	1.95E-05	0.92	1.82E-05	4.95	7.34E-05	0.12	6.51E-06
9999	2007	0.14	2.22E-05	0.92	1.82E-05	4.51	6.32E-05	0.11	6.03E-06
9999	2009	0.12	2.36E-05	0.92	1.82E-05	4.29	5.81E-05	0.11	5.79E-06
9999	2010	0.10	2.50E-05	0.92	1.82E-05	4.08	5.30E-05	0.11	5.55E-06
9999	2014	0.10	2.50E-05	0.92	1.82E-05	2.36	3.06E-05	0.06	2.78E-06
9999	2040	0.05	1.17E-05	0.92	1.82E-05	2.36	3.06E-05	0.02	1.11E-06

REFERENCES

AB 8 2X, 2009. Assembly Bill 8 2X, Chapter 6. An act to amend Section 11011 of the Government Code, to add Section 43018.2 to the Health and Safety Code, which directs ARB to amend Sections 2449.1 and 2449.2 of Title 13 of the California Code of Regulations. Approved by the Governor on February 20, 2009.

ARB, 2005. OFFROAD Modeling Change Technical Memo. Off-Road Exhaust Emissions Inventory Fuel Correction Factors., Accessed June 2010, Available at: http://www.arb.ca.gov/msei/offroad/techmemo/arb_offroad_fuels.pdf

ARB, 2007. Proposed Regulated for In-Use Off-road Diesel Vehicles, Technical Support Document, Appendix E: Emissions Inventory Methodology and Results., Accessed January 2010, Available at: http://www.arb.ca.gov/regact/2007/ordiesl07/tsdappe.pdf

ARB, 2008. Characterization of the Off-road Equipment Population, Eastern Research Group report, ARB Contract No. 04-315. Accessed June 2010 Available at: http://www.arb.ca.gov/research/apr/past/04-315.pdf

ARB, 2009a. In-Use Off-Road Diesel Vehicle Regulation Reduced Activity Records Submission; Available at: http://www.arb.ca.gov/msprog/ordiesel/documents/activityapp.pdf

ARB, 2009b. AB 8 2x Reduced activity reporting data.

ARB, 2010. Memo: Estimated Load Factors for the Showcase and Supplemental Environmental Projects (SEP) Equipment. (See Attachment C)

Baker Hughes, Accessed July 2010. Number and Location of Rotary Rigs in California. Available at: http://investor.shareholder.com/bhi/rig_counts/rc_index.cfm and http://gis.bakerhughesdirect.com/RigCounts/Attributes.aspx

Beacon Economics (Beacon), June 2010. Beaconomics: A quarter forecast for the U.S. and California from Beacon Economics. Available at: http://beaconecon.com/Beaconomics/index.php?DataSet=California|e03c_0&Region=California&Heading=California:%20Construction%20Employment

Bureau of Economic Analysis (BEA), Accessed June 2010. Regional Economic Accounts, Gross Domestic Product by State. Available at: http://bea.gov/regional/gsp/

Bureau of Labor Statistics (BLS), Accessed June 2010a. National, State and Metro Area Employment, Hours, & Earnings. Available at: http://www.bls.gov/sae/ and http://www.bls.gov/ces/

Bureau of Labor Statistics (BLS), Accessed June 2010b. Employment Projections: Occupation Employment by Industry. Available at: http://www.bls.gov/emp/ep_table_109.htm

Bureau of Transportation Statistics (BTS), Accessed June 2010a. Air Carrier Traffic Statistics. Available at: http://www.bts.gov/programs/airline_information/air_carrier_traffic_statistics/

Bureau of Transportation Statistics (BTS), Accessed June 2010b. Fuel Cost and Consumption. Available at: http://www.bts.gov/programs/airline_information/fuel_cost_and_consumption/

California State Board of Equalization (BOE), Accessed June 2010. Fuel Taxes Division Statistics & Reports. Available at: http://www.boe.ca.gov/sptaxprog/spftrpts.htm

Congressional Budget Office (CBO), January 2010. The Budget and Economic Outlook: 2010 to 2020. Available at: http://www.cbo.gov/ftpdocs/108xx/doc10871/Frontmatter.shtml

Department of Finance (DOF), 2009. Demographic Research Unit of California county-specific population estimates and projections. Available at: http://www.dof.ca.gov/research/demographic/reports/estimates/e-2/2000-09/documents/E-2_Report.xls

DOORS, 2010. Fleet records summary for April 14, 2010.

Employment Development Department (EDD), Accessed July 2010. Quarterly Census of Employment and Wages. Available at: http://www.labormarketinfo.edd.ca.gov/cgi/dataanalysis/areaselection.asp?tablename=industry

Energy Information Administration (EIA), Accessed June 2010. Topics for Petroleum Consumption/Sales. Available at: http://tonto.eia.doe.gov/dnav/pet/pet_cons_top.asp

Equipment Data Associates (EDA), 2010. UCC-1 Filings data from Catapult Database.

Federal Aviation Administration (FAA), 2010. FAA Aerospace Forecast: Fiscal Years 2010-2030. Available at: http://www.faa.gov/data_research/aviation/aerospace_forecasts/2010-2030/

Power Systems Research, 1996. See CARB Mailout#: MSC 99-32, Available at: http://www.arb.ca.gov/msprog/mailouts/mouts_99.htm

Puri A. and R. Kleinhenz, 1994. A study to develop projected activity for non-road mobile categories in California, 1970-2020. Report prepared for the California Air Resources Board, Sacramento, CA, by California State University at Fullerton, CA, ARB Contract

No. 92-732. Available at: http://www.arb.ca.gov/research/abstracts/92-732.htm

San Pedro Bay Ports, November 2009. Rubber Tire Gantry Crane Load Factor Study, Prepared by Starcrest Consulting Group. Available at: http://www.polb.com/civica/filebank/blobdload.asp?BlobID=6915

TIAX, 2003. California Public Fleet Heavy-Duty Vehicle and Equipment Inventory. Final Report. Report to California Air Resources Board., Available at: http://www.arb.ca.gov/msprog/ordiesel/documents/documents.htm

University of California Anderson School of Management (UCLA), June 2010. The UCLA Anderson Forecast for the Nation and California.

University of the Pacific Eberhardt School of Business (UOP), July 2010. California & Metro Forecast.

U.S. Census Bureau, Accessed June 2010. Total Construction Spending. Available at: http://www.census.gov/const/www/totpage.html

U.S. EPA, April 2004a. Median Life, Annual Activity, and Load Factor Values for Nonroad Engine Emissions Modeling, EPA-420-P-04-005, Report No. NR-005C. Available at: http://www.epa.gov/otaq/nonrdmdl.htm

U.S. EPA, April 2004b, Exhaust and Crankcase Emission Factors for Nonroad Engine Modeling – Compression Ignition, EPA420-P-04-009, Report No. NR-009C. Available at: http://www.epa.gov/otaq/nonrdmdl.htm