Racing Cars

By: Eric Geib and Eric Owen
Team 54
Moriarty High School

Executive Summary

In this project, we will be using C++ programming in the racing of two normal streetcars. We will be using the program to figure out the speed of the cars at any distance, the time it would take them to get to any distance, and which of the two cars would be ahead of the other at any distance. We will do this using the gear ratios, acceleration, and 0 to 100 KPH times of the cars in formulas to determine the speed of the car at any given RPM in any gear at any distance. Enjoy the read and have a nice day!

Abstract

Our project is based on racing two similar cars in the most basic of street racing ways: one on one, straight track racing. The two cars we are racing will be the 1993 Toyota Supra Turbo and 1994 Ford Saleen Mustang S-351. The problem: Who will be winning at certain distances? We began by looking online at various websites and got two very similar cars. Similar in that their ¼ mile times, 0 to 100 KPH times, top speeds and gear ratios were comparable. We will use our program to determine which car will be ahead of the other at any given distance using acceleration, gear ratios and other formulas.

Introduction

Cars are one of the most basic ways of transportation. We are out to find why two cars of the same era are so very different and what will determine the difference in the time it takes them to complete any number of distances. Using some complex formulas that we will cover later the cars will race each other to any given length and we will output this. We want to make our assumptions clear, which are: 1. The drivers of both cars are the best driver possible for that car, 2. All weather conditions are the same.

Background

There are many types of racing. The most common included the drag racing, Open-Wheel racing, Nascar racing and Baja racing. Drag racing is the most widely known and used method of racing any two cars against eachother. This type of racing is all about which car can accelerate and go the fastest. This causes a very high risk to the driver as well as the car. The amount of wind velocity against a car accelerating through space at that speed causes a great risk of the driver losing control, hence us assuming the weather is perfect and the drivers are flawless.

Problem Definition

In this project our problem is, who will be in the lead at different distances? Our code is going to race our two similar cars and output the difference between the two cars, relative to the start line, distance-wise. To solve this problem we are using equations that include distance, time, and acceleration, as well as using the Red Line RPM, 0 to 100

KPH times and the Gear Ratios to figure out how quickly each car can get to any given speed and/or distance.

Method of Solution

In this project our method of solution is as follows:

First, use the top speed and gear ratios to figure out the top speed of each car in any given gear. Second, we use this information, the cars' 0 to 60 MPH and 0 to 100 MPH to figure out how fast each car will be going at any time. Lastly, we use the information from the last step and the quarter mile time to figure out how long it would take each car to get to any distance.

Results

We have determined by the statistics we have gathered that the Supra will beat the Saleen of the line, and to 60 and 100 MPH, and the Supra will also beat the Saleen at the ¼ mile. However, because the Saleen's top speed is higher than that of the Supra, eventually the Saleen would pass the Supra and beat it. The Saleen can go faster in first gear than the Supra because the Saleen has a lower gear ratio in first gear than the Supra does. The Supra also has six gears instead of five gears like the Saleen, which explains it's speed of the line and up to 60 and 100 MPH.

Conclusions and Further Research

In later days we would like to make our program able to race several different cars at once, and make it able to include high performance car. Also, perhaps even make it able to input a car the user wants to race and race it against any car the user chooses. Basically, the conclusion of our project, or the final information, is that the Saleen loses to the Supra at shorter-distance races, but in the long run, the Saleen would pass and beat the Supra, given enough distance. These cars are very much alike in overall performance and timing, as well as gear ratios. American cars may be a bit slower than those from our Japanese neighbors in short distances, but, a word of advice: Never run from an American-made Police car in a Japanese made car, or they're catch you eventually… Our final word? If you want a fast acceleration car, buy foreign, but if you want a car that has a top speed in the rafters, buy American.

Work Sited

1). www.supercars.net
2). www.fast-autos.net
3). www.maxxskunkworks.com

Appendix A
(code)

```
/*
  Name: Fast Cars
  Author: Eric Owen and Eric Geib
  Description: Calculates 1/4 mile speeds
  Date: 24/09/02 09:19
  Copyright: E&E Inc. 2002
*/

#include <iomanip.h>
#include <iostream.h>
#include <stdlib.h>
#include <math.h>

int main()
{
float saleen;
float gear_ratio_1st;
float gear_ratio_2nd;
float gear_ratio_3rd;
float gear_ratio_4th;
float gear_ratio_5th;
float Supra;
float gear_ratio_1st_s;
float gear_ratio_2nd_s;
float gear_ratio_3rd_s;
float gear_ratio_4th_s;
float gear_ratio_5th_s;
float gear_ratio_6th_s;
float topspeed_saleen;
float topspeed_supra;
float added_gear_ratios;
float multiply_by;
float multiply_2by;
float added_gear_2ratios;
float first_gear_topspeed;
float sec_gear_topspeed;
float thrid_gear_topspeed;
float fourth_gear_topspeed;
float fifth_gear_topspeed;
float first_gear_topspeed_s;
float sec_gear_topspeed_s;
float thrid_gear_topspeed_s;
float fourth_gear_topspeed_s;
float fifth_gear_topspeed_s;
```

```
	gear_ratio_1st= 3.27;
	gear_ratio_2nd= 1.99;
	gear_ratio_3rd= 1.34;
	gear_ratio_4th= 1;
	gear_ratio_5th= .68;

	gear_ratio_1st_s= 3.83;
	gear_ratio_2nd_s= 2.36;
	gear_ratio_3rd_s= 1.69;
	gear_ratio_4th_s= 1.31;
	gear_ratio_5th_s= 1;
	gear_ratio_6th_s= .79;

	added_gear_ratios = gear_ratio_1st + gear_ratio_2nd + gear_ratio_3rd +
gear_ratio_4th + gear_ratio_5th; \\adds the gear ratios of the Saleen.
	added_gear_2ratios = gear_ratio_1st_s + gear_ratio_2nd_s + gear_ratio_3rd_s +
gear_ratio_4th_s + gear_ratio_5th_s + gear_ratio_6th_s; \\adds the gear ratios of the
Supra.

	cout<<gear_ratio_1st + gear_ratio_2nd + gear_ratio_3rd + gear_ratio_4th +
gear_ratio_5th<<" this is the gear ratio's added."<<endl; //outputs gear ratios
	cout<<gear_ratio_1st_s + gear_ratio_2nd_s + gear_ratio_3rd_s +
gear_ratio_4th_s + gear_ratio_5th_s + gear_ratio_6th_s<<" this is the gear ratio's
added."<<endl; //outputs gear ratios

	cout<<added_gear_ratios<<"added gear ratios"<<endl;
	cout<<added_gear_2ratios<<"added gear ratios 2"<<endl;

	topspeed_saleen = 170.0;  //top speeds of both cars.
	topspeed_supra = 155.0;

	multiply_by = topspeed_saleen/added_gear_ratios;  *\multiply buy to find the
gear ratios.\*
	multiply_2by = topspeed_supra/added_gear_2ratios;
	cout<<multiply_by<<"multiply by"<<endl;
	cout<<multiply_2by<<"multiply by"<<endl;

	first_gear_topspeed = multiply_by * gear_ratio_1st;
	sec_gear_topspeed = multiply_by * gear_ratio_2nd;  //finds the gear ratios for
Saleen all gears.
	thrid_gear_topspeed = multiply_by * gear_ratio_3rd;
	fourth_gear_topspeed = multiply_by * gear_ratio_4th;
	fifth_gear_topspeed = multiply_by * gear_ratio_5th;
	cout<<first_gear_topspeed<<endl;
	cout<<sec_gear_topspeed<<endl;
```

```
	cout<<thrid_gear_topspeed<<endl;
	cout<<fourth_gear_topspeed<<endl;
	cout<<fifth_gear_topspeed<<endl;

	first_gear_topspeed_s = multiply_2by * gear_ratio_1st_s;
	sec_gear_topspeed_s = multiply_2by * gear_ratio_2nd_s;  //finds the gear ratios
for the Supra all the gears.
	thrid_gear_topspeed_s = multiply_2by * gear_ratio_3rd_s;
	fourth_gear_topspeed_s = multiply_2by * gear_ratio_4th_s;
	fifth_gear_topspeed_s = multiply_2by * gear_ratio_5th_s;
	cout<<first_gear_topspeed_s<<endl;
	cout<<sec_gear_topspeed_s<<endl;
	cout<<thrid_gear_topspeed_s<<endl;
	cout<<fourth_gear_topspeed_s<<endl;
	cout<<fifth_gear_topspeed_s<<endl;




	system("PAUSE");
	system("cls");

	return 0;
	}
```