Drag Race Calculator

Software Design Document FINAL

Dawson Gomez

Date (5/1/23)

CS 225, Spring 2023

Embry-Riddle Aeronautical University

Daytona Beach campus

1 Aerospace Boulevard

Daytona Beach, FL 32114

Change Log

3/20/23

Modified formulas to be more software friendly, removed old formulas, and replaced them with modified versions of formulas found online. This should aid in accuracy of the formulas.

5/1/2023

Modified SDD to match submitted code.

**Introduction:**

The Project described in this document will be a drag race calculator, in which the program, given user inputs of weight, horsepower, drive-type, and tires, will be able to accurately calculate and simulate a drag race between two vehicles. This project will be able to accurately gage how vehicles can perform in their stock configurations, as well as how modifications such as new tires, weight reduction, or more power can affect the performance of the vehicle in a quarter mile drag race, without having to do those modifications. This will help those interested in increasing their vehicles performance on creating a road map of what modifications will be most desirable. This document contains a complete description of this project, followed by a design for the project in UML-diagram format. Following this document will allow a reader to understand, use, and recreate this project as desired.

**Problem Description:**

Drag Racing: Drag racing is a type of motor racing where two cars compete to be the first to cross a specified distance (usually 1/4th or 1/8th mile) from a standing start. The winner is the car that crosses the finish line first without committing any rule violation. The race can be conducted on a straight, flat, and paved track or a closed-off public road. In professional drag racing, vehicles are classified into different categories based on their weight, engine size, and modifications, and they race against other vehicles in the same class. Because this project is a calculator, and not a real world simulation, some factors will be left out. Assumptions will be made that the track always has the same conditions, no rules can be broken, and all drivers are the same skill. Because of this, times calculated may vary from posted manufacturer or reviewer times, this is to be expected. As safety is not a concern for the calculator, classes are irrelevant, and any vehicle can race any other vehicle.

Rules of Drag Racing:

* The race starts with a countdown, and the cars must remain stationary until the green light is signaled. (In the calculator, the system will count down from 3 to simulate this.)
* The first car to cross the finish line without committing a rule violation is the winner. (This will be printed at the time the winner crosses the finish line (for example, 12 seconds after the start if the winner has a 12 second 1/4th mile))
* The race track must be straight, flat, and paved, with clear markings indicating the start, finish, and distance markers (e.g., 60ft, 1/8th mile, and 1/4th mile).(in the calculator, this is assumed, and the calculator will print the distance markers as well as the speed they are reached as they are reached with the exception of 60ft as it is inaccurate.)
* The vehicles must meet certain safety regulations, including having a roll cage, racing harness, and fire extinguisher. (while this is a real world rule, it is irrelevant to the calculator)
* The vehicles must be classified into different categories based on their weight, engine size, and modifications, and they race against other vehicles in the same class.(again, irrelevant to the calculator)

The Drag Racing Calculator project will calculate the following metrics based on the input from the user:

* 1/4th mile time: The time taken by the vehicle to cross the 1/4th mile distance from a standing start.
* 1/8th mile time: The time taken by the vehicle to cross the 1/8th mile distance from a standing start.
* 60ft time: The time taken by the vehicle to cross the first 60ft from a standing start.
* Speed at 1/4th mile: The speed at which the vehicle crosses the 1/4th mile distance.
* Speed at 1/8th mile: The speed at which the vehicle crosses the 1/8th mile distance.

The calculations for these metrics will be based on the vehicle's acceleration and distance covered. The Drag Racing Calculator will prompt the user to input the following vehicle parameters:

* Vehicle weight(in Kilograms)
* Vehicle power (in Horse Power)
* Transmission type (Standard/Automatic)
* Tire type(normal/performance)
* Drive type(FWD/RWD)

The Drag Racing Calculator will use these parameters to calculate the vehicle's acceleration and speed at various distance markers during the drag race. The formulas used for the calculations will be based on the laws of physics, including Newton's second law of motion and the kinematic equations of motion.

The formulas for RWD vehicles are as follows:

Key:

* t = time in seconds
* v =speed in km/h
* W = weight of the vehicle in kg
* P = horsepower of the vehicle
* T = tire type factor (0 for street tires, 0.1 for drag radials, 0.2 for slicks)
* L = launch factor based on transmission type (1.57 for manual, 1.75 for automatic)
* C = 450 \* (W/1000) ^ 0.5 \* (P/400)
* K = C \* ((W^.5)/P + 0.2) ^ 4 =
* 0.7= constant factor to account for drivetrain losses (assumes rear-wheel drive, Front wheel drive will be mentioned later, they behave completely differently.)
  + 0.5 = constant factor to account for reaction time and roll-out(accounts for driver)
  + 1.55 = constant factor to estimate the 1/4th mile time based on the 1/8th mile time (assuming a typical time differential of 1.55 between the two using a Porsche 944, Chevrolet Corvette, and the VW beetle(rwd))
  + 2= constant factor to account for such a short distance in the 60ft time.

1/4th mile

t = ((W / (P \* 0.7)) ^ 0.5 \* (K / 5.825) \* (L / 1.75) \* (T + 1) + 0.5) \* 1.55

• v = 0.466 \* (60 / (time / 5.825)) ^ 1.5

1/8th mile

* t = (W / (P \* 0.7)) ^ 0.5 \* (K / 5.825) \* (L / 1.57) \* (T + 1) + 0.5
* v= 0.746 \* (60 / (time / 5.825)) ^ 1.5 \* 1.60934(accounts for increased low gear acceleration)

60ft

* t = (((W / (P \* 0.7)) ^ 0.5 \* (K / 5.825) \* (L / 1.57) \* (T + 1) + 0.5) \* 0.536) / 2

Front Wheel Drive Vehicles

In the real world, front wheel drive (FWD) vehicles behave differently from RWD vehicles. This is because due to newtons second law, as a vehicle accelerates forwards, the weight is shifted towards the rear and onto the rear tires, increasing the grip in the rear. This is why most drag cars are RWD, however for this calculator, any vehicle may be used, so addressing this difference is important. The FWD vehicles will lose grip during a launch, and have longer launch times. The formulas will all be different, having modifiers for FWD vehicles and extra formulas to calculate them. The formulas are as follows:

Key:

* t = time in seconds
* v =speed in km/h
* W = weight of the vehicle in kg
* P = horsepower of the vehicle
* T = tire type factor (0 for street tires, 0.1 for drag radials, 0.2 for slicks)
* L = launch factor based on transmission type (1.57 for manual, 1.75 for automatic)
* Constant C = 450 \* (W/1000) ^ 0.5 \* (P/400)
* Constant K = C \* ((W^.5)/P + 0.2) ^ 4
* FWD grip loss factor FWDL = (1 - exp(-0.0007 \* hp)) \* (1 - exp(-0.0002 \* weight))
* FWD Launch Factor FLF= 1.35
* FWD 60 ft factor: 1.7(to adjust for inaccuracies with small distances)

1/4th mile

* t = (K / C) ^ 0.25 \* (1 + (FWDL)) \* FLF
* v = 0.466 \* (1320(distance in ft) / (time / 0.25)) ^ 1.5

1/8th mile

* t = (K / C) ^ 0.25 \* (1 + (FWDL)) \* FLF \* (660 / 1320) ^ 0.25
* v = 0.466 \* (660 / (t / 0.125)) ^ 1.5

60ft time

* t = ((K / C) ^ 0.25 \* (1 + FWDL) \* FLF) \* FWD 60ft factor

these formulas are created by a conglomeration of knowledge from many sources, listed here, however they are based on the Porsche 944, so as a vehicle gets further from a Porsche 944’s specifications, the results may become more innacurate.

"Calculating a car's 0-60 time" by Paul Weissler, Car and Driver, September 2004

"Calculating Acceleration Times" by James Walker, The Physics Classroom

"Tire Slip, Traction, and Force" by Mark R. Warner, The Physics Classroom

"Tire Traction and Friction" by Andrew Rader, Space.com

"How to Calculate the 1/4 Mile" by Mark Gearhart, Summit Racing

"Understanding 1/8th Mile vs 1/4 Mile Drag Racing" by Mark Gearhart, Summit Racing

"Modifying Your RWD for Better Traction and Acceleration" by Mike Aguilar, OnAllCylinders

"Wheel Spin - Explained" by CarThrottle

"Vehicle dynamics and traction control" by Dean Tomazic, Society of Automotive Engineers (SAE) International

"Vehicle Performance Modeling Methodology: Vehicle Longitudinal Dynamics Model" by Jack C. McCormick, NASA Technical Memorandum

"Chassis Engineering" by Herb Adams

"Race Car Vehicle Dynamics" by William F. Milliken and Douglas L. Milliken.

**Problem Solution:**

To create the Drag Racing Calculator, a class will be implemented named CalculatorController, that calls upon VehicleCreator, which will create vehicle Objects(FWD or RWD) from a file or user input, and DragRace.DragRace will call DragRacingCalculator, which will have methods for calculating the different metrics. These will be relayed back to DragRace where they are stored for future use. The DragRace class, will “run the race” and perform the countdown, as well as printing out the times and speeds at which they occur.

The class diagram will include the following classes and their relationships:

CalculatorController: Calls upon VehicleCreator to create 2 vehicles, and DragRace

DragRacingCalculator: This class contains the methods for calculating the different metrics.

Vehicle: This will be a Generic class of vehicle, housing all of the generic vehicle information.

RWD/FWD Vehicle: This class will store the vehicle's non generic information, such as Grip modifiers and Launch factors.

DragRace: will call DragRacingCalculator and store values to print during the race.6

The CalculatorController will have the following methods:

Main(): calls other functions

The VehicleCreator will have the following methods:

createVehicle(): this method will prompt the user if they want to use a file vehicle or create their own.

The GenericVehicle class will have the following methods:

A constructor, as well as

Setters and getters for

weight: double

power: double

transmissionType: string

tireType: string

driveType: string

The FWD vehicle class will inherit all of those from the generic class, as well as include setters and getters for

FWDGripLossFactor

FWDLaunchFactor

FWD60FtFactor

The RWD vehicle class will also inherit the generic vehicle class, but have

DrivetrainLoss

Given a Vehicle, The DragRacingCalculator class will have the following methods:

calculateQuarterMileTime(): This method will use the vehicle's parameters to calculate the time taken by the vehicle to cross the 1/4th mile distance from a standing start.

calculateEighthMileTime(): This method will use the vehicle's parameters to calculate the time taken by the vehicle to cross the 1/8th mile distance from a standing start.

calculateSixtyFeetTime(): This method will use the vehicle's parameters to calculate the time taken by the vehicle to cross the first 60ft from a standing start.

calculateQuarterMileSpeed(): This method will use the vehicle's parameters to calculate the speed at which the vehicle crosses the 1/4th mile distance.

calculateEighthMileSpeed(): This method will use the vehicle's parameters to calculate the speed at which the vehicle crosses the 1/8th mile distance.

The DragRace Class shall have the following methods:

CalculateCar1(): will run and store all calculations for car 1

CalculateCar2():will run and store all calculations for car 2

Race():will perform a countdown, then print all calculated variables as they appear (if 60ft time is 2 seconds, will print after 2 seconds)

A picture containing diagram

Description automatically generated

**References:**

"Calculating a car's 0-60 time" by Paul Weissler, Car and Driver, September 2004

"Calculating Acceleration Times" by James Walker, The Physics Classroom

"Tire Slip, Traction, and Force" by Mark R. Warner, The Physics Classroom

"Tire Traction and Friction" by Andrew Rader, Space.com

"How to Calculate the 1/4 Mile" by Mark Gearhart, Summit Racing

"Understanding 1/8th Mile vs 1/4 Mile Drag Racing" by Mark Gearhart, Summit Racing

"Modifying Your RWD for Better Traction and Acceleration" by Mike Aguilar, OnAllCylinders

"Wheel Spin - Explained" by CarThrottle

"Vehicle dynamics and traction control" by Dean Tomazic, Society of Automotive Engineers (SAE) International

"Vehicle Performance Modeling Methodology: Vehicle Longitudinal Dynamics Model" by Jack C. McCormick, NASA Technical Memorandum

"Chassis Engineering" by Herb Adams

"Race Car Vehicle Dynamics" by William F. Milliken and Douglas L. Milliken.

**Appendices:**