

Physics Lecture 24 - We Finally Did It! - Real Time Race Simulator

Background/Credits

The whole involvement in Pinewood Derby Physics related to this web site began about 20 years ago. At that time the author was Research Manager in the Systems Development Department at Shell's Westhollow Research Center. His son had just lost his first pinewood derby race and the winner would not share his speed secrets. At that point, a vow was made to fully research the fundamentals of this race and hopefully never lose another one. And the vow included not keeping any secrets from anyone, especially the youngsters, whom we are supposed to be teaching. There were several PhD's who worked for me, covering a wide range of disciplines, including chemical engineering, mechanical engineering, statistics, etc. My own field is engineering physics. We formed a multi-disciplinary relationship to study pinewood derby physics, using off time such as coffee breaks, in order to review fundamentals and begin research. Although the author did 90% of the research in his lab at home, the team provided the other 10% through consultations and suggestions. A key member of the team, who still works with the author, was John H, who holds an advanced degree in physics as well. John is probably, in my opinion, one of the world's best Delphi-Pascal Windows programmers. And my son, who now works for the Woods Group as a mechanical engineer, developed this web site. The results of the 15 year research program finished in 2004 are chronologically covered, from simple to advanced, in the 526 page book shown on the Home Page. This book is called the Big Green Book (BGB) for short. Development since has focused on the race model work.

Real Time Race Simulator Development

The existing Virtual Race (VR) program, completed in 2005, applies the equations of motion developed and solved in the BGB to calculate precise times for any car/track combination. The [link](#) on home page shows as an example the equation that gives the coasting time on a horizontal run of a given length with a given input velocity. In addition, there are other equations of motion, solved for ramp exit time and velocity, for the two key types of ramps, the inclined plane ramp and the circular arc ramp. If you have your ramp height and length and coast length input, the only proviso for an actual simulation close to reality is that the real car track true and not engage in any significant center strip bumping.

Although the VR gave the expected race times as a function of the track and car parameters as input by the user, there was no feeling of an actual race. Particularly for the youngsters, what was needed was some cars going down a track. Parents/Pack Leaders needed a physics teaching tool. So a program was needed that would do the following:

- 1) Invert the solutions to the equations of motion, which in the original VR were time as a function of distance and car/track parameters, so that one had instead the down-track distance traveled as a function of time and parameters.
- 2) Use the existing VR "Race Group of Cars" program as a "Car Garage" source from which to pick any two Virtual Cars and race them side by side down a ramp plus coasting run as seen from the top.
- 3) Tie the time generator to the Windows clock, use the inverted equation solutions to convert this time into distance traveled, and input this distance to the position of a pair of small cars thus simulating their motion down a track that extends from left to right.
- 4) After the race is completed, display a graph of the separation of the two cars from start to finish, with a sensitivity much higher than the eye could see in a real race, in order to show subtle effects of physics due to car design.
- 5) Add bells and whistles to give the youngsters a better feel for actually racing, such as a "Car Body Shop" where they can change their wheels and body measurements, and a "Paint Shop" where they can call in the Windows Color Palette and choose a color for their creation.
- 6) Put a "dot" on the car that corresponds to where the builder put the center of mass in the "Body Shop" by using appropriate weights in the real car.

- 7) Install countdown lights that help a youngster practice using a stopwatch to check their time against Windows race times.
- 8) Put in a “Pause Button” that allows the user to stop the race at any point and view the car separation.
- 9) Add a “Display Results” function that will show a data table and allow the data to be copied and pasted to Excel for further study.

In addition to the general VR programming, special credit is due John H for developing and programming a method of inverting the equation of motion solutions. His method has allowed improved ramp to coast transition continuity plus a high mathematical precision that can show race car separations of only a few thousandths of an inch.

The Graphic 2-Car Race Simulator

When a user selects the VR car named MY CAR, they can change its parameters to see its race times or replace its parameters with measurements on their own car, in each case renaming the changed car and saving it. In this way a Virtual Garage of cars can be built up. We now edit the MY_CAR parameters by going to the Body Shop, as below:

Parameter	Value	Unit
Car name	MY_CAR	
Switch from cgs to mks (SI)	<input type="checkbox"/>	
Number of wheels touching track (NK)	4	
Moment of inertia of body (IC)	2635	gram cm ²
Moment of inertia of a wheel (I)	5.566	gram cm ²
Projected area of a wheel (AW)	2.817	cm ²
Projected area of body (AB)	7.405	cm ²
Drag coefficient of wheel (CW)	1.000	
Drag coefficient of body (CB)	1.000	
Mass (weight) of body plus wheels (M)	141.750	gram
Radius of a wheel (RW)	1.511	cm
Radius of an axle (RA)	0.1180	cm
Coefficient of friction (MU)	0.100	
Center of mass re body center (CM)	0.00	cm
Non-Pinewood Derby Car	<input type="checkbox"/>	
Length of car (L)	17.78	cm
Mass (weight) of wheel (MW)	3.8439196	gram

Fig 1 - MY_CAR parameters that come with the VR.

Parameter	Value	Unit
Car name	PerfCar_-7.38	
Switch from cgs to mks (SI)	<input type="checkbox"/>	
Number of wheels touching track (NK)	4	
Moment of inertia of body (IC)	0	gram cm ²
Moment of inertia of a wheel (I)	0.000	gram cm ²
Projected area of a wheel (AW)	2.817	cm ²
Projected area of body (AB)	7.405	cm ²
Drag coefficient of wheel (CW)	0.000	
Drag coefficient of body (CB)	0.000	
Mass (weight) of body plus wheels (M)	141.750	gram
Radius of a wheel (RW)	1.511	cm
Radius of an axle (RA)	0.1180	cm
Coefficient of friction (MU)	0.000	
Center of mass re body center (CM)	-7.380	cm
Non-Pinewood Derby Car	<input type="checkbox"/>	
Length of car (L)	17.78	cm
Mass (weight) of wheel (MW)	0	gram

Fig 2 - Changing to a Perfect Car with CM in body front.

From the MY_CAR parameters in the Body Shop in **Fig 1** we can create a Perfect Car by setting all moments of inertia, air drag coefficients, and friction coefficient to zero. However, we set the center of mass of this Perfect Car to be at the front axle point, a negative 7.38 centimeters from the body center. Then we rename the car PerfCar_-7.38 and save it in the Car Garage. The other PerfCar_7.38 comes with the VR program as a Standard un-editable car.

This is the fastest possible virtual car, as its CM is at the rear axles, as far to the rear as practical. Next, click on the [Race Group of Cars] tab as in **Fig 3** and select these two cars. Also select the Inclined Plane ramp (not shown) track for the race and Click [Go!] and the cars are raced for time only as in the old VR. It appears PerfCar_7.38 wins by 0.338 car lengths. But notice a new tab appears called [Graphic 2-Car Race]. Click on it to get the **Fig 4** race screen.

Run Car on Track Race Group of Cars Vary Car Parameter Graphic 2-Car Race

Check several cars and click Go! for their race times; or check just 2 and click Graphic 2-Car Race tab: **Go!** Copy to Clipboard

Car	T1	V1	T2	TT	Car Lengths
PerfCar_7.38	1.9665	487.311	0.7797	2.7462	
PerfCar_-7.38	1.9360	479.747	0.8228	2.7588	0.338

Car Garage

- PerfCar_7.38
- StdCar_1991
- MY_CAR
- PerfCar_0
- PerfCar_-7.38
- MY_CAR_1.0_1.0_1.0
- MY_CAR_0_9_9

Fig 3 - Selecting the 2 Perfect Cars to race.

On Your Mark Pause Display Pos vs Time

Track = StdTrk_HOUSTON_I_BT Paint Shop
 Car-A = PerfCar_7.38 Car A
 Car-B = PerfCar_-7.38 Car B

Car-B Lengths Behind

0.00

0.0

Down Track Distance of Car-A

Fig 4 - The Graphic 2-Car Race Screen. Dots mark the CM positions of the cars. [Click on the video below.](#)

On Your Mark Pause Display Pos vs Time

Track = StdTrk_HOUSTON_I_BT Paint Shop
 Car-A = PerfCar_7.38 Car A
 Car-B = PerfCar_-7.38 Car B

Car-B Lengths Behind

0.00

0.0

Down Track Distance of Car-A

In the video, the user is shown clicking on the new Perfect Car_-7.38 and choosing a bright green color. The Standard Car remains white and is one of two Standard Cars which can't be painted. The user then clicks [On Your Mark], the countdown timer flashes 3 colors in sequence and the race begins, lasting precisely 2.7462 seconds (fastest car) as obtained earlier. Notice the race finish times show up at the end of the car name at finish time. In the screen capture video at 20 fps car movement may not appear 100% smooth compared with races on a real computer. Remember a lot of calculations are going on during the race. On a modern 19" LCD monitor with at least 1280 horizontal screen resolution, the race track action is smooth, especially if other high CPU usage programs are not running.

Pausing the race when the cars are at the end of ramp (EoR) as in **Fig 5** we see that the cars are perfectly tied. The teaching point here is that no matter where you place a car's CM on an inclined plane ramp, its angle of travel, and hence its acceleration, is the same all the way down the ramp.

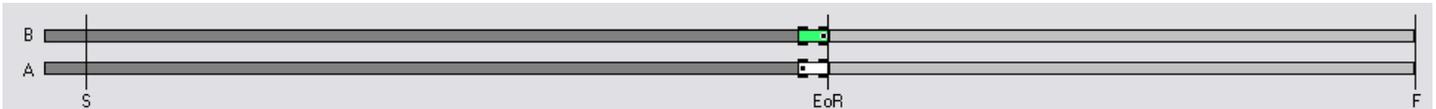


Fig 5 - The white/green race paused at the EoR. Here the cars have zero separation and are tied.

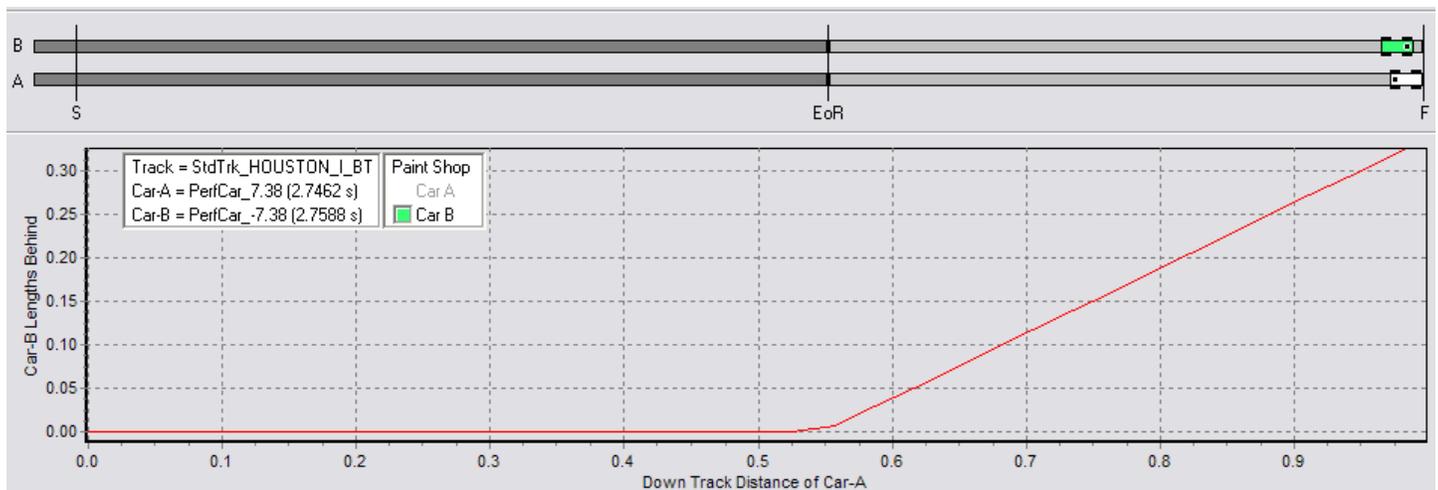


Fig 6 - The white/green race finish. Notice the separation graph shows white's extra velocity boost at the EoR compared to the green car caused it to pull away on the coasting run. The finish times appear at the end of the car name.

Continuing the teaching point, we see in **Fig 6** the end of the race. What happened was that at the EoR, the green car's acceleration has stopped, as its CM is now beginning to coast, sort of pulling the rest of the body as a weightless trailer behind it. But look at the white car. It's CM still has almost a full body length of ramp acceleration left, so it's still gaining velocity as it pushes its weightless front end towards the finish line. The red separation line tells the whole story, as one can see that the extra velocity "boost" of the white car causes it to pull away steadily, crossing the finish line at about a third (0.338) of a car length ahead of the green perfect car.

The inclined plane track used in the above example is similar to the commercial Best Track, specified in some detail in **Lecture 20**. But you should see what happens when we race the green and white cars on a track that has a circular arc shaped ramp. These tracks, like the original track in the Cub Leader's "How To" book, or the Micro Wizard tracks, allow the ramp to "sag" in the center, approaching a shape close to the arc of a circle. You may be in for a surprise, but you'll have to try it and see for yourself.

This lecture is getting pretty long, so we'll finish it with a final example. First, consider that a car has 12 parameters that can be independently varied. So the number of different cars that can be generated is extremely large, and so is the number of different pairs that can be selected to race. We will choose as an example the circular arc ramp we just mentioned, and we will choose 2 slightly different versions of MY_CAR as shown earlier in **Fig 1**.

Take MY_CAR into the car Body Shop again and start work. It will take only about 30 seconds to change parameters, rename, and save. Play like one car builder was in a hurry so he just used the block directly with several sharp square corners and wheels that weren't rounded and he knew his air drag coefficients would be high at about 1.0. But to compensate, he thought he would just move his center of mass weights back about 1 centimeter so that his CM measured 1.0. He let his son paint the car who blended up a nice olive drab color because he had designed the car to be like an army jeep. The other Dad helped his son sand the body sharp corners and also slightly round the sharp edges of the wheel tread (still leaving it flat enough to pass inspection). He had read somewhere that drag coefficients could be reduced about 10% by simple edge rounding. They were also in a hurry and just left the center of mass in the body center at $CM = 0$. His son felt that red was an appropriate color to paint the car. When the cars were saved to the garage, the olive drab car was named MY_CAR_1.0_1.0_1.0 to remind the owners of the CM, body drag coefficient, and wheel drag coefficient respectively. The red car was named MY_CAR_0.9_9 for the same reason. Which car would you think would win?

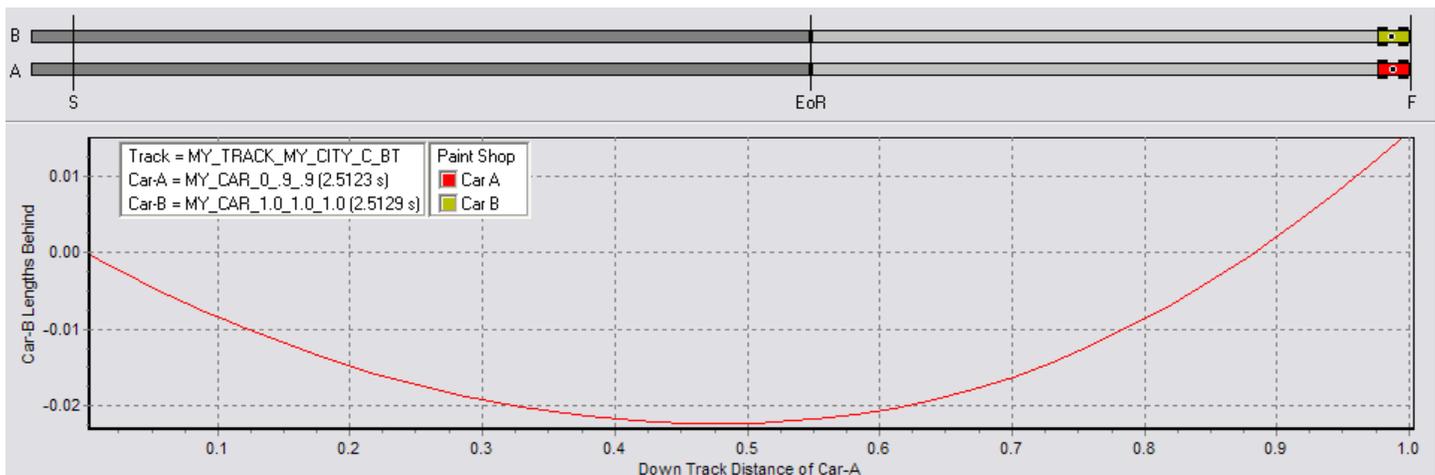


Fig 7 - The red/olive drab race finish. Notice the separation graph shows olive's CM placement gave it an early advantage from the start until the speed built up. As the velocity increased, the lower drag coefficient of the red car became the dominant factor.

The teaching point here is that the separation graph shows olive's more rearward CM placement gave it an early advantage from the start until the speed built up. As the velocity increased, the lower air drag coefficient of the red car became the dominant factor and red continued to gain on the coasting run winning by a nose (about a tenth of an inch) at the finish. Had we run the perfect green and white cars on the circular arc ramped track, we could have again isolated the CM effect and not have the several other factors that slow a car down be involved. Likewise, one can build a virtual perfect car and add any single factor like wheel moment of inertia to study only its effect throughout the race. Another teaching point comes from observing this race on an inclined plane ramp. A figure showing this along with brief text is on the web site under the [Graphic 2-Car Race](#) page.

Remember the equations of motion do not include random effects like caused by center strip bumping. But they are a good predictor of behavior for cars designed and tuned to track straight on a good level track. Even if extraneous effects are still present, if indeed random they will average out, and the average will approach the predictions from the equations of motion.

As a final point it should be mentioned that the tracks do not have to be of standard dimensions. For example, a coasting run could be hundreds of feet long. Then the dark gray ramp part where acceleration occurs would be rather short. This is because the program is set up to scale the overall ramp plus coast length from zero to one. The cars, however, stay the same size on the screen, but appear to move slower in real time, roughly in proportion to the apparent elevation from which the track is viewed. Nevertheless, the movement relative to the front bumper of the car and the car separation is still controlled by the equations of motion. If a car slows to zero velocity before reaching the finish line, the race will be aborted with an error signal.