Introduction to the PhORS articles By Brian Beckman

"I started this series in 1991 for my local racing club's printed newsletter. The web had just been born, though the Internet was not yet public. Nonetheless, I distributed the articles over the Internet at that time and they become reasonably well known, especially amongst the autocrossing community in the US. The first 13 parts were written in 1991, so they contain some very dated ideas, such as using Scheme for writing simulations. However, the entire series is presented here, as originally written. Perhaps at some later time I will consolidate and update the series, but for now, I am focusing on writing new parts. There are currently a number of 'live threads' in the discussion that I wish to pursue at length.

My overall goal with the series is to present a fresh outlook on racing physics, understandable to the technically inclined non-specialist. The problems I consider come from a variety of sources. Often, they're motivated by computer simulation, and just as often they arise from competition experiences. Some of the later articles get very technical, but I always try to balance conceptual discussion, which everyone should be able to understand, with mathematical analysis, which might of interest only to specialists, and with numerical results, which, again, should be universally accessible.

When I first started the series, I purposely avoided the standard reference sources, preferring to figure things out myself from first principles. In the past ten years, a number of superior source books, papers, and programs have become available, and it is no longer sensible for me to avoid them. I've had my fun, now it is time to 'get real.' So, in the later articles, I refer to the well known books by Milliken, Gillespie, Genta, and Carroll Smith; as well as to free simulation packages such as RARS, TORCS, and Racer.

There is a tremendous amount of activity in racing simulation nowadays that computer hardware is fast enough to permit extremely detailed modelling of racing cars in real time. The realism of Grand-Prix Legends, for instance, was unimaginable in real time in 1991. Despite this growth, I continue to hope that the Physics of Racing series can fulfil its original dual roles of translating racing lore and craft into hardcore physics and of making that physics understandable to real-world working race drivers and teams.

Finally, I wish to point out that these articles are FREE. I retain the copyright ONLY to prevent the kind of theft that would make the articles difficult to copy, meaning that I grant to everyone, everywhere a perpetual, transferable, universal, royalty-free license to copy, host, post, translate, convert, transform, and reproduce the articles in any form whatever, asking only that the content and attribution not be changed and that the rights of anyone, anywhere to further copy the articles not be restricted, say, by charging money for copies."

The Physics of Racing, Part 1: Weight Transfer

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

Most autocrossers and race drivers learn early in their careers the importance of balancing a car. Learning to do it consistently and automatically is one essential part of becoming a truly good driver. While the skills for balancing a car are commonly taught in drivers' schools, the rationale behind them is not usually adequately explained. That rationale comes from simple physics. Understanding the physics of driving not only helps one be a better driver, but increases one's enjoyment of driving as well. If you know the deep reasons why you ought to do certain things you will remember the things better and move faster toward complete internalisation of the skills.

Balancing a car is controlling weight transfer using throttle, brakes, and steering. This article explains the physics of weight transfer. You will often hear instructors and drivers say that applying the brakes shifts weight to the front of a car and can induce oversteer. Likewise, accelerating shifts weight to the rear, inducing understeer, and cornering shifts weight to the opposite side, unloading the inside tyres. But why does weight shift during these manoeuvres? How can weight shift when everything is in the car bolted in and strapped down? Briefly, the reason is that inertia acts through the centre of gravity (CG) of the car, which is above the ground, but adhesive forces act at ground level through the tyre contact patches. The effects of weight transfer are proportional to the height of the CG off the ground. A flatter car, one with a lower CG, handles better and quicker because weight transfer is not so drastic as it is in a high car.

The rest of this article explains how inertia and adhesive forces give rise to weight transfer through Newton's laws. The article begins with the elements and works up to some simple equations that you can use to calculate weight transfer in any car knowing only the wheelbase, the height of the CG, the static weight distribution, and the track, or distance between the tyres across the car. These numbers are reported in shop manuals and most journalistic reviews of cars.

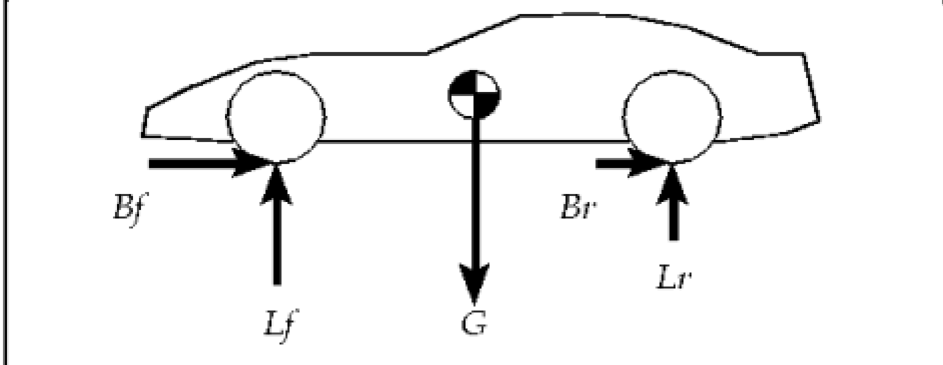
Most people remember Newton's laws from school physics. These are fundamental laws that apply to all large things in the universe, such as cars. In the context of our racing application, they are:

The first law: **a car in straight-line motion at a constant speed will keep such motion until acted on by an external force**. The only reason a car in neutral will not coast forever is that friction, an external force, gradually slows the car down. Friction comes from the tyres on the ground and the air flowing over the car. The tendency of a car to keep moving the way it is moving is the inertia of the car, and this tendency is concentrated at the CG point.

The second law: **When a force is applied to a car, the change in motion is proportional to the force divided by the mass of the car**. This law is expressed by the famous equation $\boldsymbol{F = ma}$, where $\boldsymbol{F}$ is a force, $\boldsymbol{m}$ is the mass of the car, and $\boldsymbol{a}$ is the acceleration, or change in motion, of the car. A larger force causes quicker changes in motion, and a heavier car reacts more slowly to forces. Newton's second law explains why quick cars are powerful and lightweight. The more $\boldsymbol{F}$ and the less $\boldsymbol{m}$ you have, the more $\boldsymbol{a}$ you can get.

The third law: **Every force on a car by another object, such as the ground, is matched by an equal and opposite force on the object by the car**. When you apply the brakes, you cause the tyres to push forward against the ground, and the ground pushes back. As long as the tyres stay on the car, the ground pushing on them slows the car down.

Let us continue analysing braking. Weight transfer during accelerating and cornering are mere variations on the theme. We won't consider subtleties such as suspension and tyre deflection yet. These effects are very important, but secondary. The figure shows a car and the forces on it during a "one g" braking manoeuvre. One g means that the total braking force equals the weight of the car, say, in pounds.



A car braking at 1 "gee"

In this figure, the black and white "pie plate" in the centre is the CG. $\boldsymbol{G}$ is the force of gravity that pulls the car toward the centre of the Earth. This is the weight of the car; weight is just another word for the force of gravity. It is a fact of Nature, only fully explained by Albert Einstein, that gravitational forces act through the CG of an object, just like inertia. This fact can be explained at deeper levels, but such an explanation would take us too far off the subject of weight transfer.

L_f is the lift force exerted by the ground on the front tyre, and L_r is the lift force on the rear tyre. These lift forces are as real as the ones that keep an airplane in the air, and they keep the car from falling through the ground to the centre of the Earth.

We don't often notice the forces that the ground exerts on objects because they are so ordinary, but they are at the essence of car dynamics. The reason is that the magnitude of these forces determine the ability of a tyre to stick, and imbalances between the front and rear lift forces account for understeer and oversteer. The figure only shows forces on the car, not forces on the ground and the CG of the Earth. Newton's third law requires that these equal and opposite forces exist, but we are only concerned about how the ground and the Earth's gravity affect the car.

If the car were standing still or coasting, and its weight distribution were 50-50, then L_f would be the same as L_r. It is always the case that L_f plus L_r equals G, the weight of the car. Why? Because of Newton's first law. The car is not changing its motion in the vertical direction, at least as long as it doesn't get airborne, so the total sum of all forces in the vertical direction must be zero. G points down and counteracts the sum of L_f and L_r, which point up.

Braking causes L_f to be greater than L_r. Literally, the "rear end gets light," as one often hears racers say. Consider the front and rear braking forces, B_f and B_r, in the diagram. They push backwards on the tyres, which push on the wheels, which push on the suspension parts, which push on the rest of the car, slowing it down. But these forces are acting at ground level, not at the level of the CG. The braking forces are indirectly slowing down the car by pushing at ground level, while the inertia of the car is 'trying' to keep it moving forward as a unit at the CG level.

The braking forces create a rotating tendency, or torque, about the CG. Imagine pulling a tablecloth out from under some glasses and candelabra. These objects would have a tendency to tip or rotate over, and the tendency is greater for taller objects and is greater the harder you pull on the cloth. The rotational tendency of a car under braking is due to identical physics.

The braking torque acts in such a way as to put the car up on its nose. Since the car does not actually go up on its nose (we hope), some other forces must be counteracting that tendency, by Newton's first law. G cannot be doing it since it passes right through the centre of gravity. The only forces that can counteract that tendency are the lift forces, and the only way they can do so is for L_f to become greater than L_r. Literally, the ground pushes up harder on the front tyres during braking to try to keep the car from tipping forward.

By how much does L_f exceed L_r? The braking torque is proportional to the sum of the braking forces and to the height of the CG. Let's say that height is 20 inches. The counterbalancing torque resisting the braking torque is proportional to L_f and half the wheelbase (in a car with 50-50 weight distribution), minus L_r times half the wheelbase since L_r is helping the braking forces upend the car. L_f has a lot of work to do: it must resist the torques of both the braking forces and the lift on the rear tyres. Let's say the wheelbase is 100 inches. Since we are braking at one g, the braking forces equal G, say, 3200 pounds. All this is summarized in the following equations:

$$3200 \text{ lbs times } 20 \text{ inches} = L_f \text{ times } 50 \text{ inches} - L_r \text{ times } 50 \text{ inches}$$

$$L_f + L_r = 3200 \text{ lbs (this is always true)}$$

With the help of a little algebra, we can find out that

$$L_f = 1600 + 3200 / 5 = 2240 \text{ lbs}$$

$$L_r = 1600 - 3200 / 5 = 960 \text{ lbs}$$

Thus, by braking at one g in our example car, we add 640 pounds of load to the front tyres and take 640 pounds off the rears! This is very pronounced weight transfer.

By doing a similar analysis for a more general car with CG height of h, wheelbase w, weight G, static weight distribution d expressed as a fraction of weight in the front, and braking with force B, we can show that

$$L_f = dG + Bh / w$$

$$L_r = (1 - d)G - Bh / w$$

These equations can be used to calculate weight transfer during acceleration by treating acceleration force as negative braking force. If you have acceleration figures in gees, say from a *G-analyst* or other device, just multiply them by the weight of the car to get acceleration forces (Newton's second law!). Weight transfer during cornering can be analysed in a similar way, where the track of the car replaces the wheelbase and d is always 50% (unless you account for the weight of the driver). Those of you with science or engineering backgrounds may enjoy deriving these equations for yourselves. The equations for a car doing a combination of braking and cornering, as in a trail-braking manoeuvre, are much more complicated and require some mathematical tricks to derive.

Now you know why weight transfer happens. The next topic that comes to mind is the physics of tyre adhesion, which explains how weight transfer can lead to understeer and oversteer conditions.

The Physics of Racing, Part 2: Keeping Your Tyres Stuck to the Ground

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

In last month's article, we explained the physics behind weight transfer. That is, we explained why braking shifts weight to the front of the car, accelerating shifts weight to the rear, and cornering shifts weight to the outside of a curve. Weight transfer is a side-effect of the tyres keeping the car from flipping over during manoeuvres. We found out that a one *g* braking manoeuvre in our 3200 pound example car causes 640 pounds to transfer from the rear tyres to the front tyres. The explanations were given directly in terms of Newton's fundamental laws of Nature.

This month, we investigate what causes tyres to stay stuck and what causes them to break away and slide. We will find out that you can make a tyre slide either by pushing too hard on it or by causing weight to transfer off the tyre by your control inputs of throttle, brakes, and steering. Conversely, you can cause a sliding tyre to stick again by pushing less hard on it or by transferring weight to it. The rest of this article explains all this in term of (you guessed it) physics.

This knowledge, coupled with a good "instinct" for weight transfer, can help a driver predict the consequences of all his or her actions and develop good instincts for staying out of trouble, getting out of trouble when it comes, and driving consistently at ten tenths. It is said of Tazio Nuvolari, one of the greatest racing drivers ever, that he knew at all times while driving the weight on each of the four tyres to within a few pounds. He could think, while driving, how the loads would change if he lifted off the throttle or turned the wheel a little more, for example. His knowledge of the physics of racing enabled him to make tiny, accurate adjustments to suit every circumstance, and perhaps to make these adjustments better than his competitors. Of course, he had a very fast brain and phenomenal reflexes, too.

I am going to ask you to do a few physics "lab" experiments with me to investigate tyre adhesion. You can actually do them, or you can just follow along in your imagination. First, get a tyre and wheel off your car. If you are a serious autocrosser, you probably have a few loose sets in your garage. You can do the experiments with a heavy box or some object that is easier to handle than a tyre, but the numbers you get won't apply directly to tyres, although the principles we investigate will apply.

Weigh yourself both holding the wheel and not holding it on a bathroom scale. The difference is the weight of the tyre and wheel assembly. In my case, it is 50 pounds (it would be a lot less if I had those $3000 Jongbloed wheels! Any sponsors reading?). Now put the wheel on the ground or on a table and push sideways with your hand against the tyre until it slides. When you push it, push down low near the point where the tyre touches the ground so it doesn't tip over.

The question is, how hard did you have to push to make the tyre slide? You can find out by putting the bathroom scale between your hand and the tyre when you push. This procedure doesn't give a very accurate reading of the force you need to make the tyre slide, but it gives a rough estimate. In my case, on the concrete walkway in front of my house, I had to push with 85 pounds of force (my neighbours don't bother staring at me any more; they're used to my strange antics). On my linoleum kitchen floor, I only had to push with 60 pounds (but my wife does stare at me when I do this stuff in the house). What do these numbers mean?

They mean that, on concrete, my tyre gave me 85 / 50 = 1.70 gees of sideways resistance before sliding. On a linoleum race course (ahem!), I would only be able to get 60 / 50 = 1.20$\boldsymbol{g}$. We have directly experienced the physics of grip with our bare hands. The fact that the tyre resists sliding, up to a point, is called the *grip phenomenon*. If you could view the interface between the ground and the tyre with a microscope, you would see complex interactions between long-chain rubber molecules bending, stretching, and locking into concrete molecules creating the grip. Tyre researchers look into the detailed workings of tyres at these levels of detail.

Now, I'm not getting too excited about being able to achieve 1.70$\boldsymbol{g}$ cornering in an autocross. Before I performed this experiment, I frankly expected to see a number below 1$\boldsymbol{g}$. This rather unbelievable number of 1.70$\boldsymbol{g}$ would certainly not be attainable under driving conditions, but is still a testimony to the rather unbelievable state of tyre technology nowadays. Thirty years ago, engineers believed that one $\boldsymbol{g}$ was theoretically impossible from a tyre. This had all kinds of consequences. It implied, for example, that dragsters could not possibly go faster than 200 miles per hour in a quarter mile: you can go $\sqrt{2\boldsymbol{ax}}$ = 198.48 mph if you can keep 1$\boldsymbol{g}$ acceleration all the way down the track. Nowadays, drag racing safety watchdogs are working hard to keep the cars under 300 mph; top fuel dragsters launch at more than 3 gees.

For the second experiment, try weighing down your tyre with some ballast. I used a couple of dumbbells slung through the centre of the wheel with rope to give me a total weight of 90 pounds. Now, I had to push with 150 pounds of force to move the tyre sideways on concrete. Still about 1.70$\boldsymbol{g}$. We observe the fundamental law of adhesion: the force required to slide a tyre is proportional to the weight supported by the tyre. When your tyre is on the car, weighed down with the car, you cannot push it sideways simply because you can't push hard enough.

The force required to slide a tyre is called the *adhesive limit* of the tyre, or sometimes the *stiction*, which is a slang combination of "stick" and "friction." This law, in mathematical form, is

$$\boldsymbol{F} \leq \boldsymbol{\mu W}$$

where $\boldsymbol{F}$ is the force with which the tyre resists sliding; $\boldsymbol{\mu}$ is the *coefficient of static friction* or *coefficient of adhesion*; and $\boldsymbol{W}$ is the weight or vertical load on the tyre contact patch. Both $\boldsymbol{F}$ and $\boldsymbol{W}$ have the units of force (remember that weight is the force of gravity), so $\boldsymbol{\mu}$ is just a number, a proportionality constant. This equation states that the sideways force a tyre can withstand before sliding is less than or equal to $\boldsymbol{\mu}$ times $\boldsymbol{W}$. Thus, $\boldsymbol{\mu W}$ is the maximum sideways force the tyre can withstand and is equal to the stiction. We often like to speak of the sideways acceleration the car can achieve, and we can convert the stiction force into acceleration in gees by dividing by $\boldsymbol{W}$, the weight of the car $\boldsymbol{\mu}$ can thus be measured in gees.

The coefficient of static friction is not exactly a constant. Under driving conditions, many effects come into play that reduce the stiction of a good autocross tyre to somewhere around 1.10$\boldsymbol{g}$. These effects are deflection of the tyre, suspension movement, temperature, inflation pressure, and so on. But the proportionality law still holds reasonably true under these conditions. Now you can see that if you are cornering, braking, or accelerating at the limit, which means at the adhesive limit of the tyres, any weight transfer will cause the tyres unloaded by the weight transfer to pass from sticking into sliding.

Actually, the transition from sticking ‘mode’ to sliding mode should not be very abrupt in a well-designed tyre. When one speaks of a “forgiving” tyre, one means a tyre that breaks away slowly as it gets more and more force or less and less weight, giving the driver time to correct. Old, hard tyres are, generally speaking, less forgiving than new, soft tyres. Low-profile tyres are less forgiving than high-profile tyres. Slicks are less forgiving than DOT tyres. But these are very broad generalities and tyres must be judged individually, usually by getting some word-of-mouth recommendations or just by trying them out in an autocross. Some tyres are so unforgiving that they break away virtually without warning, leading to driver dramatics usually resulting in a spin. Forgiving tyres are much easier to control and much more fun to drive with.

“Driving by the seat of your pants,” means sensing the slight changes in cornering, braking, and acceleration forces that signal that one or more tyres are about to slide. You can sense these change literally in your seat, but you can also feel changes in steering resistance and in the sounds the tyres make. Generally, tyres ‘squeak’ when they are nearing the limit, ‘squeal’ at the limit, and ‘squall’ over the limit. I find tyre sounds very informative and always listen to them while driving.

So, to keep your tyres stuck to the ground, be aware that accelerating gives the front tyres less stiction and the rear tyres more, that braking gives the front tyre more stiction and the rear tyres less, and that cornering gives the inside tyres less stiction and the outside tyres more. These facts are due to the combination of weight transfer and the grip phenomenon. Finally, drive smoothly, that is, translate your awareness into gentle control inputs that always keep appropriate tyres stuck at the right times. This is the essential knowledge required for car control, and, of course, is much easier said than done. Later articles will use the knowledge we have accumulated so far to explain understeer, oversteer, and chassis set-up.

The Physics of Racing, Part 3: Basic Calculations

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

In the last two articles, we plunged right into some relatively complex issues, namely weight transfer and tyre adhesion. This month, we regroup and review some of the basic units and dimensions needed to do dynamical calculations. Eventually, we can work up to equations sufficient for a full-blown computer simulation of car dynamics. The equations can then be 'doctored' so that the computer simulation will run fast enough to be the core of an autocross computer game. Eventually, we might direct this series of articles to show how to build such a game in a typical microcomputer programming language such as C or BASIC, or perhaps even my personal favourite, LISP. All of this is in keeping with the spirit of the series, the Physics of Racing, because so much of physics today involves computing. Software design and programming are essential skills of the modern physicist, so much so that many of us become involved in computing full time.

Physics is the science of measurement. Perhaps you have heard of highly abstract branches of physics such as quantum mechanics and relativity, in which exotic mathematics is in the forefront. But when theories are taken to the laboratory (or the race course) for testing, all the mathematics must boil down to quantities that can be measured. In racing, the fundamental quantities are distance, time, and mass. This month, we will review basic equations that will enable you to do quick calculations in your head while cooling off between runs. It is very valuable to develop a skill for estimating quantities quickly, and I will show you how.

Equations that don't involve mass are called *kinematic*. The first kinematic equation relates speed, time, and distance. If a car is moving at a constant speed or velocity, $\boldsymbol{v}$, then the distance $\boldsymbol{d}$ it travels in time $\boldsymbol{t}$ is

$$\boldsymbol{d} = \boldsymbol{v}\boldsymbol{t}$$

or velocity times time. This equation really expresses nothing more than the definition of velocity.

If we are to do mental calculations, the first hurdle we must jump comes from the fact that we usually measure speed in miles per hour (mph), but distance in feet and time in seconds. So, we must modify our equation with a conversion factor, like this

$$\boldsymbol{d}\,(\text{feet}) = \boldsymbol{v}\,\frac{\text{miles}}{\text{hour}}\,\boldsymbol{t}\,(\text{seconds})\,\frac{5280\ \text{ft/mile}}{3600\ \text{seconds/hour}}$$

If you "cancel out" the units parts of this equation, you will see that you get feet on both the left and right hand sides, as is appropriate, since equality is required of any equation. The conversion factor is 5280/3600, which happens to equal 22/15. Let's do a few quick examples. How far does a car go in one second (remember, say, "one-one-thousand, two-one-thousand," *etc.* to yourself to count off seconds)? At fifteen mph, we can see that we go

$$\boldsymbol{d} = 15 \text{ mph times } 1 \text{ sec times } 22/15 = 22 \text{ feet}$$

or about 1 and a half car lengths for a 14 and 2/3 foot car like a late-model Corvette. So, at 30 mph, a second is three car lengths and at 60 mph it is six. If you lose an autocross by 1 second (and you'll be pretty good if you can do that with all the good drivers in our region), you're losing by somewhere between 3 and 6 car lengths! This is because the average speed in an autocross is between 30 and 60 mph.

Every time you plough a little or get a little sideways, just visualize your competition overtaking you by a car length or so. One of the reasons autocross is such a difficult sport, but also such a pure sport, from the driver's standpoint, is that you can't make up this time. If you blow a corner in a road race, you may have a few laps in which to make it up. But to win an autocross against good competition, you must drive nearly perfectly. The driver who makes the fewest mistakes usually wins!

The next kinematic equation involves acceleration. It so happens that the distance covered by a car at constant acceleration from a standing start is given by

$$\boldsymbol{d} = \frac{1}{2}\boldsymbol{a}\boldsymbol{t}^2$$

or 1/2 times the acceleration times the time, squared. What conversions will help us do mental calculations with this equation? Usually, we like to measure acceleration in ***g***s. One ***g*** happens to be 32.1 feet per second squared. Fortunately, we don't have to deal with miles and hours here, so our equation becomes,

$$\boldsymbol{d}\,(\text{feet}) = 16\boldsymbol{a}\,(\boldsymbol{g}\text{s})\,\boldsymbol{t}\,(\text{seconds})^2$$

roughly. So, a car accelerating from a standing start at ½***g***, which is a typical number for a good, stock sports car, will go 8 feet in 1 second. Not very far! However, this picks up rapidly. In two seconds, the car will go 32 feet, or over two car lengths.

Just to prove to you that this isn't crazy, let's answer the question "How long will it take a car accelerating at ½***g*** to do the quarter mile?" We invert the equation above (recall your high school algebra), to get

$$\boldsymbol{t} = \sqrt{\boldsymbol{d}(\text{feet})\,16\boldsymbol{a}\,(\boldsymbol{gs})}$$

and we plug in the numbers: the quarter mile equals 1320 feet, $\boldsymbol{a} = ½\boldsymbol{g}$, and we get $\boldsymbol{t} = \sqrt{1320/8} = \sqrt{165}$ which is about 13 seconds. Not too unreasonable! A real car will not be able to keep up full ½***g*** acceleration for a quarter mile due to air resistance and reduced torque in the higher gears. This explains why real (stock) sports cars do the quarter mile in 14 or 15 seconds.

The more interesting result is the fact that it takes a full second to go the first 8 feet. So, we can see that the launch is critical in an autocross. With excessive wheel spin, which robs you of acceleration, you can lose a whole second right at the start. Just visualize your competition pulling 8 feet ahead instantly, and that margin grows because they are 'hooked up' better.

For doing these mental calculations, it is helpful to memorize a few squares. 8 squared is 64, 10 squared is 100, 11 squared is 121, 12 squared is 144, 13 squared is 169, and so on. You can then estimate square roots in your head with acceptable precision.

Finally, let's examine how engine torque becomes force at the drive wheels and finally acceleration. For this examination, we will need to know the mass of the car. Any equation in physics that involves mass is called *dynamic*, as opposed to kinematic. Let's say we have a Corvette that weighs 3200 pounds and produces 330 foot-pounds of torque at the crankshaft. The Corvette's automatic transmission has a first gear ratio of 3.06 (the auto is the trick set up for 'vettes-just ask Roger Johnson or Mark Thornton). A transmission is nothing but a set of circular, rotating levers, and the gear ratio is the leverage, multiplying the torque of the engine. So, at the output of the transmission, we have

$$3.06 \text{ x } 330 = 1010 \text{ foot-pounds}$$

of torque. The differential is a further lever-multiplier, in the case of the Corvette by a factor of 3.07, yielding 3100 foot pounds at the centre of the rear wheels (this is a lot of torque!). The distance from the centre of the wheel to the ground is about 13 inches, or 1.08 feet, so the maximum force that the engine can put to the ground in a rearward direction (causing the ground to push back forward-remember part 1 of this series!) in first gear is

$$3100 \text{ foot-pounds} / 1.08 \text{ feet} = 2870 \text{ pounds}$$

Now, at rest, the car has about 50/50 weight distribution, so there is about 1600 pounds of load on the rear tyres. You will remember from last month's article on tyre adhesion that the tyres cannot respond with a forward force much greater than the weight that is on them, so they simply will spin if you stomp on the throttle, asking them to give you 2870 pounds of force.

We can now see why it is important to squeeeeeeeze the throttle gently when launching. In the very first instant of a launch, your goal as a driver is to get the engine up to where it is pushing on the tyre contact patch at about 1600 pounds. The tyres will squeal or hiss just a little when you get this right. Not so coincidentally, this will give you a forward force of about 1600 pounds, for an $\boldsymbol{F = ma}$ (part 1) acceleration of about ½***g***, or half the weight of the car. The main reason a car will accelerate with only ½***g*** to start with is that half of the weight is on the front wheels and is unavailable to increase the stiction of the rear, driving tyres. Immediately, however, there will be some weight transfer to the rear. Remembering part 1 of this series again, you can estimate that about 320 pounds will be transferred to the rear immediately. You can now ask the tyres to give you a little more, and you can gently push on the throttle. Within a second or so, you can be at full throttle, putting all that torque to work for a beautiful hole shot!

In a rear drive car, weight transfer acts to make the driving wheels capable of withstanding greater forward loads. In a front drive car, weight transfer works against acceleration, so you have to be even more gentle on the throttle if you have a lot of power. An all-wheel drive car puts all the wheels to work delivering force to the ground and is theoretically the best.

Technical people call this style of calculating “back of the envelope,” which is a somewhat picturesque reference to the habit we have of writing equations and numbers on any piece of paper that happens to be handy. You do it without calculators or slide rules or abacuses. You do it in the garage or the pits. It is not exactly precise, but gives you a rough idea, say within 10 or 20 percent, of the forces and accelerations at work. And now you know how to do back-of-the-envelope calculations, too.

The Physics of Racing, Part 4: There Is No Such Thing as Centrifugal Force

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

One often hears of "centrifugal force." This is the apparent force that throws you to the outside of a turn during cornering. If there is anything loose in the car, it will immediately slide to the right in a left hand turn, and *vice versa*. Perhaps you have experienced what happened to me once. I had omitted to remove an empty Pepsi can hidden under the passenger seat. During a particularly aggressive run (something for which I am not unknown), this can came loose, fluttered around the cockpit for a while, and eventually flew out the passenger window in the middle of a hard left hand corner.

I shall attempt to convince you, in this month's article, that centrifugal force is a fiction, and a consequence of the fact first noticed just over three hundred years ago by Newton that objects tend to continue moving in a straight line unless acted on by an external force.

When you turn the steering wheel, you are trying to get the front tyres to push a little sideways on the ground, which then pushes back, by Newton's third law. When the ground pushes back, it causes a little sideways acceleration. This sideways acceleration is a change in the sideways velocity. The acceleration is proportional to the sideways force, and inversely proportional to the mass of the car, by Newton's second law. The sideways acceleration thus causes the car to veer a little sideways, which is what you wanted when you turned the wheel. If you keep the steering and throttle at constant positions, you will continue to go mostly forwards and a little sideways until you end up where you started. In other words, you will go in a circle. When driving through a sweeper, you are going part way around a circle. If you take skid pad lessons (highly recommended), you will go around in circles all day.

If you turn the steering wheel a little more, you will go in a tighter circle, and the sideways force needed to keep you going is greater. If you go around the same circle but faster, the necessary force is greater. If you try to go around too fast, the adhesive limit of the tyres will be exceeded, they will slide, and you will not stick to the circular path-you will not "make it."

From the discussion above, we can see that in order to turn right, for example, a force, pointing to the right, must act on the car that veers it away from the straight line it naturally tries to follow. If the force stays constant, the car will go in a circle. From the point of view of the car, the force always points to the right. From a point of view outside the car, at rest with respect to the ground, however, the force points toward the centre of the circle. From this point of view, although the force is constant in *magnitude*, it changes *direction*, going around and around as the car turns, always pointing at the geometrical centre of the circle. This force is called *centripetal*, from the Greek for "centre seeking." The point of view on the ground is privileged, since objects at rest from this point of view feel no net forces. Physicists call this special point of view an *inertial frame of reference*. The forces measured in an inertial frame are, in a sense, more correct than those measured by a physicist riding in the car. Forces measured inside the car are *biased* by the centripetal force.

Inside the car, all objects, such as the driver, feel the natural inertial tendency to continue moving in a straight line. The driver receives a centripetal force from the car through the seat and the belts. If you don't have good restraints, you may find yourself pushing with your knee against the door and tugging on the controls in order to get the centripetal force you need to go in a circle with the car. It took me a long time to overcome the habit of tugging on the car in order to stay put in it. I used to come home with bruises on my left knee from pushing hard against the door during an autocross. I found that a tight five-point harness helped me to overcome this unnecessary habit. With it, I no longer think about body position while driving - I can concentrate on trying to be smooth and fast. As a result, I use the wheel and the gearshift lever for steering and shifting rather than for helping me stay put in the car!

The 'forces' that the driver and other objects inside the car feel are actually centripetal. The term *centrifugal*, or "centre fleeing," refers to the inertial tendency to resist the centripetal force and to continue going straight. If the centripetal force is constant in magnitude, the centrifugal tendency will be constant. There is no such thing as centrifugal force (although it is a convenient fiction for the purpose of some calculations).

Let's figure out exactly how much sideways acceleration is needed to keep a car going at speed $\boldsymbol{v}$ in a circle of radius $\boldsymbol{r}$. We can then convert this into force using Newton's second law, and then figure out how fast we can go in a circle before exceeding the adhesive limit-in other words, we can derive maximum cornering speed. For the following discussion, it will be helpful for you to draw little back-of-the-envelope pictures (I'm leaving them out, giving our editor a rest from transcribing my graphics into the newsletter).

Consider a very short interval of time, far less than a second. Call it $\boldsymbol{dt}$ ($\boldsymbol{d}$ stands for "delta," a Greek letter mathematicians use as shorthand for "tiny increment"). In time $\boldsymbol{dt}$, let us say we go forward a distance $\boldsymbol{dx}$ and sideways a distance $\boldsymbol{ds}$. The forward component of the velocity of the car is approximately $\boldsymbol{v} = \boldsymbol{dx} / \boldsymbol{dt}$. At the beginning of the time interval $\boldsymbol{dt}$, the car has no sideways velocity. At the end, it has sideways velocity $\boldsymbol{ds} / \boldsymbol{dt}$. In the time $\boldsymbol{dt}$, the car has thus had a change in sideways velocity of $\boldsymbol{ds} / \boldsymbol{dt}$. Acceleration is, precisely, the change in velocity over a certain time, divided by the time; just as velocity is the change in position over a certain time, divided by the time. Thus, the sideways acceleration is

$$a = \frac{ds}{dt}\frac{1}{dt}$$

How is ***ds*** related to ***r***, the radius of the circle? If we go forward by a fraction ***f*** of the radius of the circle, we must go sideways by exactly the same fraction of ***dx*** to stay on the circle. This means that ***ds = f dx***. The fraction ***f*** is, however, nothing but ***dx / r***. By this reasoning, we get the relation

$$ds = dx\frac{dx}{r}$$

We can substitute this expression for ***ds*** into the expression for ***a***, and remembering that $\boldsymbol{v} = \boldsymbol{dx} / \boldsymbol{dt}$, we get the final result

$$a = \frac{ds}{dt}\frac{1}{dt} = \frac{dx}{dt}\frac{1}{r} = \frac{v^2}{r}$$

This equation simply says quantitatively what we wrote before: that the acceleration (and the force) needed to keep to a circular line increases with the velocity and increases as the radius gets smaller.

What was *not* appreciated before we went through this derivation is that the necessary acceleration increases as the *square* of the velocity. This means that the centripetal force your tyres must give you for you to make it through a sweeper is very sensitive to your speed. If you go just a little bit too fast, you might as well go *much* too fast - you're not going to make it. The following table shows the maximum speed that can be achieved in turns of various radii for various sideways accelerations. This table shows the value of the expression

$$\frac{15}{22}\sqrt{32.1\boldsymbol{a}(\text{gees})\,\boldsymbol{r}(\text{feet})}$$

which is the solution of $\boldsymbol{a} = \boldsymbol{v}^2/\boldsymbol{r}$ for $\boldsymbol{v}$, the velocity. The conversion factor 15/22 converts $\boldsymbol{v}$ from feet per second to miles per hour, and 32.1 converts ***a*** from gees to feet per second squared. We covered these conversion factors in part 3 of this series.

Table 1. Speed (Miles Per Hour)

Acceleration	Radius (Feet)				
(Gees)	**50.00**	**100.0**	**150.0**	**200.0**	**500.0**
0.25	13.66	19.31	23.66	27.32	43.19
0.50	19.31	27.32	33.45	38.63	61.08
0.75	23.66	33.45	40.97	47.31	74.81
1.00	27.32	38.63	47.31	54.63	86.38
1.25	30.54	43.19	52.90	61.08	96.57
1.50	33.45	47.31	57.94	66.91	105.79
1.75	36.13	51.10	62.59	72.27	114.27
2.00	38.63	54.63	66.91	77.26	122.16

For autocrossing, the columns for 50 and 100 feet and the row for 1.00***g*** are most germane. The table tells us that to achieve 1.00***g*** sideways acceleration in a corner of 50 foot radius (this kind of corner is all too common in autocross), a driver must not go faster than 27.32 miles per hour. To go 30 mph, 1.25***g*** is required, which is probably not within the capability of an autocross tyre at this speed. There is not much subjective difference between 27 and 30 mph, but the objective difference is usually between making a controlled run and spinning badly.

The absolute fastest way to go through a corner is to be just over the limit near the exit, in a controlled slide. To do this, however, you must be pointed in just such a way that when the car breaks loose and slides to the exit of the corner it will be pointed straight down the optimal racing line at the exit when it "hooks up" again. You can smoothly add throttle during this manoeuvre and be really moving out of the corner. But you must do it smoothly. It takes a long time to learn this, and probably a lifetime to perfect it, but it feels absolutely triumphal when done right. I have not figured out how to drive through a sweeper, except for the exit, at anything greater than the limiting velocity because sweepers are just too long to slide around. If anyone (Ayrton Senna, perhaps?) knows how, please tell me!

The chain of reasoning we have just gone through was first discovered by Newton and Leibniz, working independently. It is, in fact, a derivation in differential calculus, the mathematics of very small quantities. Newton keeps popping up. He was perhaps the greatest of all physicists, having discovered the laws of motion, the law of gravity, and calculus, among other things such as the fact that white light is made up of multiple colours mixed together.

It is an excellent diagnostic exercise to drive a car around a circle marked with cones or chalk and gently to increase the speed until the car slides. If the front breaks away first, your car has natural understeer, and if the rear slides first, it has natural oversteer. You can use this information for chassis tuning. Of course, this is only to be done in safe circumstances, on a rented skid pad or your own private parking lot. The police will gleefully give you a ticket if they catch you doing this in the wrong places.

The Physics of Racing, Part 5: Introduction to the Racing Line

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

This month, we analyse the best way to go through a corner. "Best" means in the least time, at the greatest average speed. We ask "what is the shape of the driving line through the corner that gives the best time?" and "what are the times for some other lines, say hugging the outside or the inside of the corner?" Given the answers to these questions, we go on to ask "what shape does a corner have to be before the driving line I choose doesn't make any time difference?" The answer is a little surprising.

The analysis presented here is the simplest I could come up with, and yet is still quite complicated. My calculations went through about thirty steps before I got the answer. Don't worry, I won't drag you through the mathematics; I just sketch out the analysis, trying to focus on the basic principles. Anyone who would read through thirty formulas would probably just as soon derive them for him or herself.

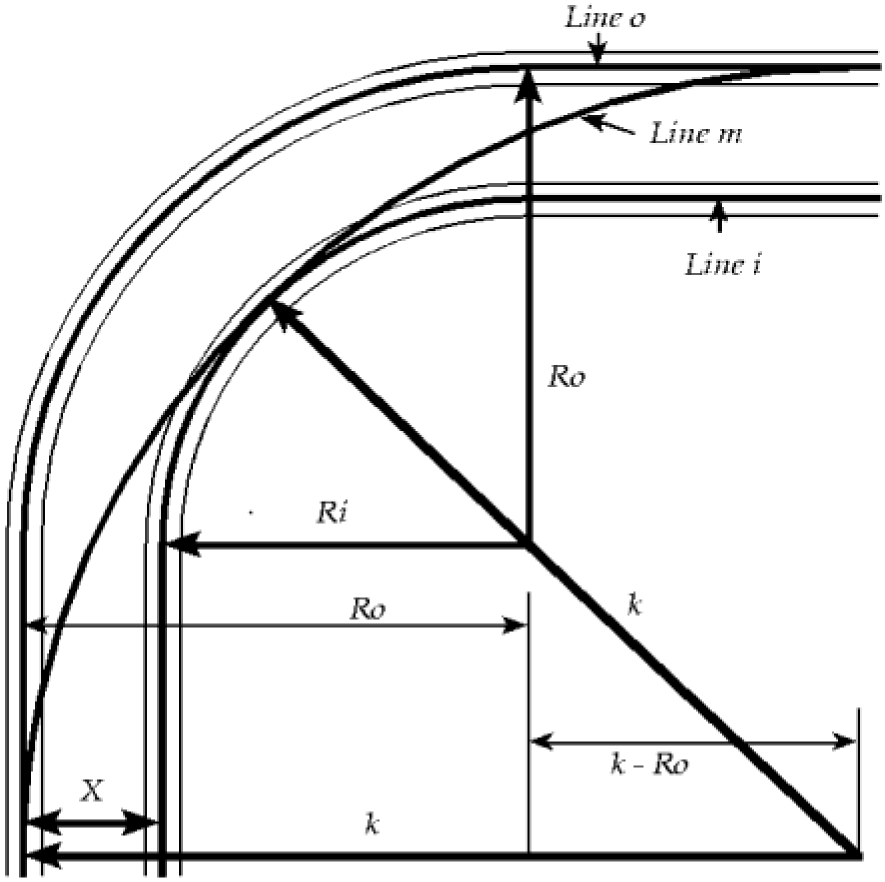
There are several simplifying assumptions I make to get through the analysis. First of all, I consider the corner in isolation; as an abstract entity lifted out of the rest of a course. The actual best driving line through a corner depends on what comes before it and after it. You usually want to optimise exit speed if the corner leads onto a straight. You might not apex if another corner is coming up. You may be forced into an unfavourable entrance by a prior curve or slalom.

Speaking of road courses, you will hear drivers say things like "you have to do such-and-such in turn six to be on line for turn ten and the front straight." In other words, actions in any one spot carry consequences pretty much all the way around. The ultimate drivers figure out the line for the entire course and drive it as a unit, taking a Zen-like approach. When learning, it is probably best to start out optimising each kind of corner in isolation, then work up to combinations of two corners, three corners, and so on. In my own driving, there are certain kinds of three corner combinations I know, but mostly I work in twos. I have a long way to go.

It is not feasible to analyse an actual course in an exact, mathematical way. In other words, although science can provide general principles and hints, finding the line is, in practice, an art. For me, it is one of the most fun parts of racing.

Other simplifying assumptions I make are that the car can either accelerate, brake, or corner at constant speed, with abrupt transitions between behaviours. Thus, the lines I analyse are splices of accelerating, braking, and cornering phases. A real car can, must, and should do these things in combination and with smooth transitions between phases. It is, in fact, possible to do an exact, mathematical analysis with a more realistic car that transitions smoothly, but it is much more difficult than the splice-type analysis and does not provide enough more quantitative insight to justify its extra complexity for this article.

Our corner is the following ninety-degree right-hander:



This figure actually represents a family of corners with any constant width, any radius, and short straights before and after. First, we go through the entire analysis with a particular corner of 75 foot radius and 30 foot width, then we end up with times for corners of various radii and widths.

Let us define the following parameters:
$\boldsymbol{r}$ = radius of corner centre line = 75 feet
$\boldsymbol{W}$ = width of course = 30 feet
$\boldsymbol{r_o}$ = radius of outer edge = $\boldsymbol{r} + \frac{1}{2}\boldsymbol{W}$ = 90 feet
$\boldsymbol{r_i}$ = radius of inner edge = $\boldsymbol{r} - \frac{1}{2}\boldsymbol{W}$ = 60 feet

Now, when we drive this corner, we must keep the tyres on the course, otherwise we get a lot of cone penalties (or go into the weeds). It is easiest (though not so realistic) to do the analysis considering the path of the centre of gravity of the car rather than the paths of each wheel. So, we define an *effective* course, narrower than the real course, down which we may drive the centre of the car.

ω = width of car = 6 feet

R_o = effective outer radius = $r_o - \frac{1}{2}\omega$ = 87 feet

R_i = effective inner radius = $r_i + \frac{1}{2}\omega$ = 63 feet

X = effective width of course = $W - \omega$ = 24 feet

This course is indicated by the labels and the thick radius lines in the figure.

From last month's article, we know that for a fixed centripetal acceleration, the maximum driving speed increases as the square root of the radius. So, if we drive the largest possible circle through the effective corner, starting at the outside of the entrance straight, going all the way to the inside in the middle of the corner (the *apex*), and ending up at the outside of the exit straight, we can corner at the maximum speed. Such a line is shown in the figure as the thick circle labelled "line ***m***." This is a simplified version of the classic racing line through the corner. Line ***m*** reaches the apex at the geometrical centre of the circle, whereas the classic racing line reaches an apex after the geometrical centre – a *late* apex – because it assumes we are accelerating out of the corner and must therefore have a continuously increasing radius in the second half and a slightly tighter radius in the first half to prepare for the acceleration. But, we continue analysing the geometrically perfect line because it is relatively easy. The figure shows also Line ***i***, the *inside* line, which come up the inside of the entrance straight, corners on the inside, and goes down the inside of the exit straight; and Line ***o***, the *outside* line, which comes up the outside, corners on the outside, and exits on the outside.

One might argue that there are certain advantages of line ***i*** over line ***m***. Line ***i*** is considerably shorter than Line ***m***, and although we have to go slower through the corner part, we have less total distance to cover and might get through faster. Also, we can accelerate on part of the entrance chute and all the way on the exit chute, while we have to drive line ***m*** at constant speed. Let's find out how much time it takes to get through lines ***i*** and ***m***. We include line ***o*** for completeness, even though it looks bad because it is both slower and longer than ***m***.

If we assume a maximum centripetal acceleration of 1.10g, which is just within the capability of autocross tyres, we get the following speeds for the cornering phases of Lines ***i***, ***o***, and ***m***:

Cornering Speed (mph)		
Line ***i***	Line ***o***	Line ***m***
32.16	37.79	48.78
v_i	v_o	v_m

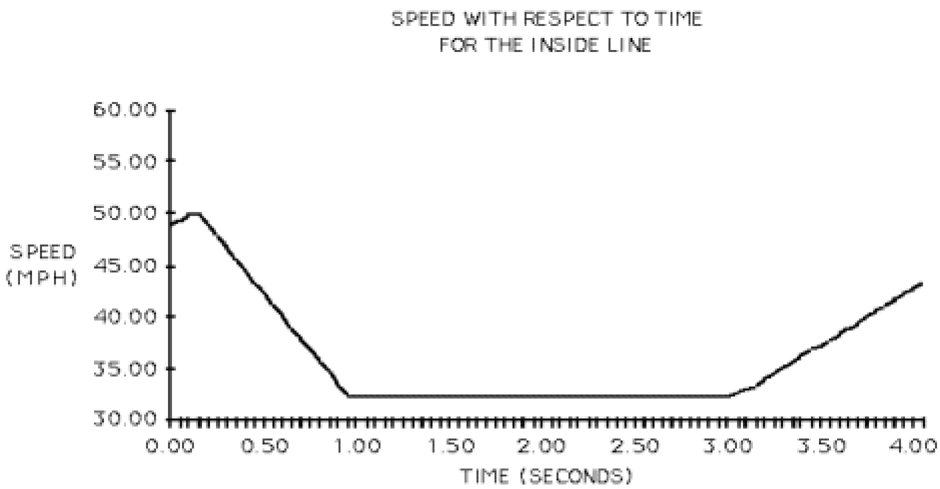
Line ***m*** is all cornering, so we can easily calculate the time to drive it once we know the radius, labelled ***k*** in the figure. A geometrical analysis results in

$$k = 3.414(R_o - 0.707R_i) = 145 \text{ feet}$$

and the time is

$$t_m = \left(\frac{\pi}{2}k\right) \Big/ \left(\frac{22}{15}v_m\right) = 3.18 \text{ seconds}$$

For line ***i***, we accelerate for a bit, brake until we reach 32.16 mph, corner at that speed, and then accelerate on the exit. Let's assume, to keep the comparison fair, that we have timing lights at the beginning and end of line ***m*** and that we can begin driving line ***i*** at 48.78 mph, the same speed that we can drive line ***m***. Let us also assume that the car can accelerate at ½***g*** and brake at 1***g***. Our driving plan for line ***i*** results in the following velocity profile:



Because we can begin by accelerating, we start beating line ***m*** a little. We have to brake hard to make the corner. Finally, although we accelerate on the exit, we don't quite come up to 48.78 mph, the exit speed for line ***m***. But, we don't care about exit speed, only time through the corner. Using the velocity profile above, we can calculate the time for line ***i***, call it t_i, to be 4.08 seconds. Line ***i*** loses by 9/10ths of a second. It is a fair margin to lose an autocross by this much over a whole course, but this analysis shows we can lose it in just one typical corner! In this case, line ***i*** is a catastrophic mistake. Incidentally, line ***o*** takes 4.24 seconds = t_o.

What if the corner were tighter or of greater radius? The following table shows some times for 30 foot wide corners of various radii:

radius	**30.00**	**45.00**	**60.00**	**75.00**	**90.00**	**95.00**
t_o	3.99	4.06	4.15	4.24	4.35	4.38
t_i	3.94	3.94	4.00	4.08	4.17	4.21
t_m	2.64	2.83	3.01	3.18	3.34	3.39
margin	1.30	1.11	1.01	0.90	0.83	0.82

Line ***i*** *never* beats line ***m*** even though that as the radius increases, the margin of loss decreases. The trend is intuitive because corners of greater radius are also longer and the extra speed in line ***m*** over line ***i*** is less. The margin is greatest for tight corners because the width is a greater fraction of the length and the speed differential is greater.

How about for various widths? The following table shows times for a 75 foot radius corner of several widths:

width	**10.00**	**30.00**	**50.00**	**70.00**	**90.00**
t_o	2.68	4.24	5.47	6.50	7.41
t_i	2.62	4.08	5.32	6.45	7.51
t_m	2.46	3.18	3.77	4.27	4.73
margin	0.16	0.90	1.55	2.18	2.79

The wider the course, the greater the margin of loss. This is, again, intuitive since on a wide course, line ***m*** is a really large circle through even a very tight corner. Note that line ***o*** becomes better than line ***i*** for wide courses. This is because the speed differential between lines ***o*** and ***i*** is very great for wide courses. The most notable fact is that line ***m*** beats line ***i*** by 0.16 seconds even on a course that is only four feet wider than the car! You really must "use up the whole course."

So, the answer is, under the assumptions made, that the inside line is *never* better than the classic racing line. For the splice-type car behaviour assumed, I conjecture that *no* line is faster than line ***m***.

We have gone through a simplified kind of *variational analysis*. Variational analysis is used in all branches of physics, especially mechanics and optics. It is possible, in fact, to express all theories of physics, even the most arcane, in variational form, and many physicists find this form very appealing. It is also possible to use variational analysis to write a computer program that finds an approximately perfect line through a complete, realistic course.

The Physics of Racing, Part 6: Speed and Horsepower

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

The title of this month's article consists of two words dear to every racer's heart. This month, we do some "back of the envelope" calculations to investigate the basic physics of speed and horsepower (the "back of the envelope" style of calculating was covered in part 3 of this series).

How much horsepower does it take to go a certain speed? At first blush, a physicist might be tempted to say "none," because he or she remembers Newton's first law, by which an object moving at a constant speed in a straight line continues so moving forever, even to the end of the Universe, unless acted on by an external force. Everyone knows, however, that it is necessary to keep your foot on the gas to keep a car moving at a constant speed. Keeping your foot on the gas means that you are making the engine apply a backward force to the ground, which applies a reaction force forward on the car, to keep the car moving. In fact, we know a few numbers from our car's shop manual. A late model Corvette, for example, has a top speed of about 150 miles per hour and about 240 hp. This means that if you keep your foot *all* the way down, using up all 240 hp, you can eventually go 150 mph. It takes a while to get there. In this car, you can get to 60 mph in about 6 seconds (if you don't spin the drive wheels), to 100 mph in about 15 seconds, and 150 in about a minute.

All this seems to contradict Newton's first law. What is going on? An automobile moving at constant speed in a straight line on level ground is, in fact, acted on by a number of external forces that tend to slow it down. Without these forces, the car *would* coast forever as guaranteed by Newton's first law. You must counteract these forces with the engine, which indirectly creates a reaction force that keeps the car going. When the car is going at a constant speed, the *net* force on the car, that is, the speeding-up forces minus the slowing-down forces, is zero.

The most important external, slowing-down force is *air resistance* or *drag*. The second most important force is friction between the tyres and the ground, the so-called *rolling resistance*. Both these forces are called *resistance* because they always act to oppose the forward motion of the car in whatever direction it is going. Another physical effect that slows a car down is internal friction in the drive train and wheel bearings. Acting internally, these forces cannot slow the car. However, they push backwards on the tyres, which push forward on the ground, which pushes back by Newton's third law, slowing the car down. The internal friction forces are opposed by external reaction forces, which act as slight braking forces, slowing the car. So, Newton and the Universe are safe; everything is working as it should.

How big are the resistance forces, and what role does horsepower play? The physics of air resistance is very complex and an area of vigorous research today. Most of this research is done by the aerospace industry, which is technologically very closely related to the automobile industry, especially when it comes to racing. We'll slog through some arithmetic here to come up with a table that shows how much horsepower it takes to sustain speed. Those who don't have the stomach to go through the math can skim the next few paragraphs.

We cannot derive equations for air resistance here. We'll just look them up. My source is *Fluid Mechanics*, by L. D. Landau and E. M. Lifshitz, two eminent Russian physicists. They give the following approximate formula:

$$\boldsymbol{F} = \tfrac{1}{2}\boldsymbol{Cd\rho v}^2$$

The factors in this equation are the following:

$\boldsymbol{Cd}$ =coefficient of friction, a factor depending on the shape of a car and determined by experiment; for a late model Corvette it is about 0.30;

$\boldsymbol{A}$ =frontal area of the car; for a Corvette, it is about 20 square feet;

$\boldsymbol{\rho}$ =Greek letter *rho*, density of air, which we calculate below;

$\boldsymbol{v}$ =speed of the car.

Let us calculate the density of air using "back of the envelope" methods. We know that air is about 79% Nitrogen and 21% Oxygen. We can look up the fact that Nitrogen has a molecular weight of about 28 and Oxygen has a molecular weight of about 32. What is molecular weight? It is the mass (not the weight, despite the name) of 22.4 litres of gas. It is a number of historical convention, just like feet and inches, and doesn't have any real science behind it. So, we figure that air has an average molecular weight of

$$\frac{79\%\,\text{of}\ 28 + 21\%\,\text{of}\ 32 = 28.84\,\text{grams}}{22.4\,\text{litres}} = 1.29\,\text{gm/l}$$

I admit to using a calculator to do this calculation, against the spirit of the "back of the envelope" style. So sue me.

We need to convert 1.29 gm/l to pounds of mass per cubic foot so that we can do the force calculations in familiar, if not convenient, units. It is worthwhile to note, as an aside, that a great deal of the difficulty of doing calculations in the physics of racing has to do with the traditional units of feet, miles, and pounds we use. The metric system makes all such calculations vastly simpler. Napoleon Bonaparte wanted to convert the world the metric system (mostly so his own soldiers could do artillery calculations quickly in their heads) but it is still not in common use in America nearly 200 years later!

Again, we look up the conversion factors. My source is *Engineering Formulas* by Kurt Gieck, but they can be looked up in almost any encyclopaedia or dictionary. There are 1000 litres in a cubic meter, which in turn contains 35.51 cubic feet. Also, a pound-mass contains 453.6 grams. These figures give us, for the density of air

$$1.29\frac{\text{gm}}{\text{litre}}\frac{\text{lb}-\text{mass}}{453.6\,\text{gm}}\frac{1000\,\text{litres}}{1\,\text{metre}^3}\frac{1\,\text{metre}^3}{35.51\,\text{ft}^3}=0.0801\frac{\text{lb - mass}}{\text{ft}^3}$$

This says that a cubic foot of air weighs 8 hundredths of a pound, and so it does! Air is much more massive than it seems, until you are moving quickly through it, that is.

Let's finish off our equation for air resistance. We want to fill in all the numbers except for speed, $\boldsymbol{v}$, using the Corvette as an example car so that we can calculate the force of air resistance for a variety of speeds. We get

$$\boldsymbol{F}=\tfrac{1}{2}\boldsymbol{(}0.30=\boldsymbol{Cd}\,\boldsymbol{)(}20\text{ft}^2=\boldsymbol{A}\,\boldsymbol{)(}0.080\frac{\text{lb - mass}}{\text{ft}^3}=\boldsymbol{\rho}\,\boldsymbol{)}\boldsymbol{v}^2=0.24\boldsymbol{v}^2\,\frac{\text{lb - mass}}{\text{ft}}$$

We want, at the end, to have $\boldsymbol{v}$ in miles per hour, but we need in feet per seconds for the calculations to come out right. We recall that there are 22 feet per second for every 15 miles per hour, giving us

$$\begin{aligned}\boldsymbol{F}&=0.24\left(\frac{22\,\text{ft/sec}}{15\,\text{mph}}\boldsymbol{v}\,(\text{mph})\right)^2\frac{\text{lb - mass}}{\text{ft}}\\&=0.517\,(\boldsymbol{v}\,(\text{mph}))^2\,\frac{\text{lb - mass ft}}{\text{sec}^2}\end{aligned}$$

Now (this gets confusing, and it wouldn't be if we were using the metric system), a pound mass is a phoney unit. A lb-mass is concocted to have a weight of 1 pound under the action of the Earth's gravity. Pounds are a unit of force or weight, not of mass. We want our force of air resistance in pounds of force, so we have to divide lb-mass ft/sec^2 by 32.1, numerically equal to the acceleration of Earth's gravity in ft/sec^2, to get pounds of force. You just have to know these things. This was a lot of work, but it's over now. We finally get

$$\boldsymbol{F}=\frac{0.517}{32.1}(\boldsymbol{v}\,(\text{mph}))^2=0.0161\,(\boldsymbol{v}\,(\text{mph}))^2\,\text{pounds}$$

Let's calculate a few numbers. The following table gives the force of air resistance for a number of interesting speeds:

$\boldsymbol{v}$ (mph)	**15**	**30**	**60**	**90**	**120**	**150**
$\boldsymbol{F}$ (pounds)	3.60	14.5	58.0	130	232	362

We can see that the force of air resistance goes up rapidly with speed, until we need over 350 pounds of constant force just to overcome drag at 150 miles per hour. We can now show where horsepower comes in.

Horsepower is a measure of *power*, which is a technical term in physics. It measures the amount of work that a force does as it acts over time. *Work* is another technical term in physics. It measures the actual effect of a force in moving an object over a distance. If we move an object one foot by applying a force of one pound, we are said to be doing one foot-pound of work. If it takes us one second to move the object, we have exerted one foot-pound per second of power. A horsepower is 550 foot-pounds per second. It is another one of those historical units that Napoleon hated and that has no reasonable origin in science.

We can expend one horsepower by exerting 550 pounds of force to move an object 1 foot in 1 second, or by exerting 1 pound of force to move an object 550 feet in 1 second, or by exerting 1 pound of force to move an object 1 foot in 0.001818 seconds, and so on. All these actions take the same amount of power. Incidentally, a horsepower happens to be equal also to 745 watts. So, if you burn about 8 light bulbs in your house, someone somewhere is expending at least one horsepower (and probably more like four or five) in electrical forces to keep all that going for you, and you pay for the service at the end of the month!.

All this means that to find out how much horsepower it takes to overcome air resistance at any speed, we need to multiply the force of air resistance by speed (in feet per second, converted from miles per hour), and divide by 550, to convert foot-lb/sec to horsepower. The formula is

$$\boldsymbol{P} = \boldsymbol{F}\boldsymbol{v} = \frac{0.0161}{550}\frac{22}{15}\boldsymbol{v}^3 = \frac{0.354}{8250}(\boldsymbol{v}\,(\text{mph}))^3 \text{ horsepower}$$

and we get the following numbers from the formula for a few interesting speeds.

$\boldsymbol{v}$ (mph)	**30**	**55**	**65**	**90**	**120**	**150**	**200**
$\boldsymbol{F}$ (pounds)	14.5	48.7	68.0	130	232	362	644
horsepower	1.16	7.14	11.8	31.3	74.2	145	344

I put 55 mph and 65 mph in this table to show why some people think that the 55 mph national speed limit saves gasoline. It only requires about 7 hp to overcome drag at 55 mph, while it requires almost 12 hp to overcome drag at 65. Fuel consumption is approximately proportional to horsepower expended.

More interesting to the racer is the fact that it takes 145 hp to overcome drag at 150 mph. We know that our Corvette example car has about 240 hp, so about 95 hp must be going into overcoming rolling resistance and the slight braking forces arising from internal friction in the drive train and wheel bearings. Race cars capable of going 200 mph usually have at least 650 hp, about 350 of which goes into overcoming air resistance. It is probably possible to go 200 mph with a car in the 450-500 hp range, but such a car would have very good aerodynamics; expensive, low-friction internal parts; and low rolling resistance tyres, which are designed to have the smallest possible contact patch like high performance bicycle tyres, and are therefore not good for handling.

The Physics of Racing, Part 7: The Traction Budget

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

This month, we introduce the traction budget. This is a way of thinking about the traction available for car control under various conditions. It can help you make decisions about driving style, the right line around a course, and diagnosing handling problems. We introduce a diagramming technique for visualizing the traction budget and combine this with a well-known visualization tool, the "circle of traction," also known as the circle of friction. So this month's article is about tools, conceptual and visual, for thinking about some aspects of the physics of racing.

To introduce the traction budget, we first need to visualize a tyre in contact with the ground. Figure 1 shows how the bottom surface of a tyre might look if we could see that surface by looking down from above. In other words, this figure shows an imaginary "X-ray" view of the bottom surface of a tyre. For the rest of the discussion, we will always imagine that we view the tyre this way. From this point of view, "up" on the diagram corresponds to forward forces and motion of the tyre and the car, "down" corresponds to backward forces and motion, "left" corresponds to leftward forces and motion, and "right" on the diagram corresponds to rightward forces and motion.

The bottom surface of a tyre viewed from the top as though with "X-ray vision."

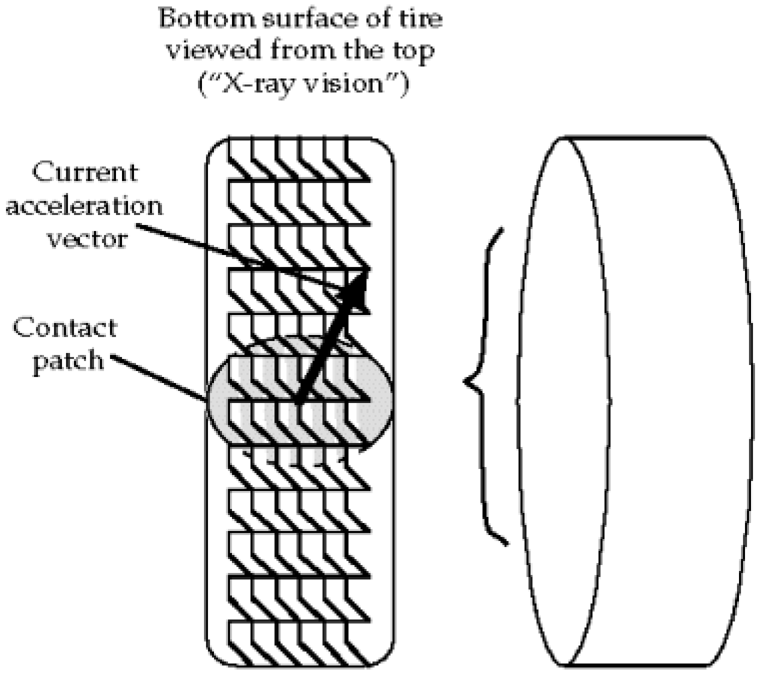


Figure 1: The bottom surface of a tire viewed from the top as though with "X ray vision."

The figure shows a shaded, elliptical region, where the tyre presses against the ground. All the interaction between the tyre and the ground takes place in this *contact patch*: that part of the tyre that touches the ground. As the tyre rolls, one bunch of tyre molecules after another move into the contact patch. But the patch itself more-or-less keeps the same shape, size, and position relative to the axis of rotation of the tyre and the car as a whole. We can use this fact to develop a simplified view of the interaction between tyre and ground. This simplified view let's us quickly and easily do approximate calculations good within a few percent. (A full-blown, mathematical analysis requires tyre coordinates that roll with the tyre, ground coordinates fixed on the ground, car coordinates fixed to the car, and many complicated equations relating these coordinate systems; the last few percent of accuracy in a mathematical model of tyre-ground interaction involves a great deal more complexity.)

You will recall that forces on the tyre from the ground are required to make a car change either its speed of motion or its direction of motion. Thinking of the X-ray vision picture, forces pointing up are required to make the car accelerate, forces pointing down are required to make it brake, and forces pointing right and left are required to make the car turn. Consider forward acceleration, for a moment. The engine applies a torque to the axle. This torque becomes a force, pointing backwards (down, on the diagram), that the tyre applies to the ground. By Newton's third law, the ground applies an equal and opposite force, therefore pointing forward (up), on the contact patch. This force is transmitted back to the car, accelerating it forward. It is easy to get confused with all this backward and forward action and reaction. Remember to think only about the forces on the tyre and to ignore the forces on the ground, which point the opposite way.

You will also recall that a tyre has a limited ability to stick to the ground. Apply a force that is too large, and the tyre slides. The maximum force that a tyre can take depends on the weight applied to the tyre: $\boldsymbol{F \leq \mu W}$ where $\boldsymbol{F}$ is the force on the tyre, $\boldsymbol{\mu}$ is the coefficient of adhesion (and depends on tyre compound, ground characteristics, temperature, humidity, phase of the moon, *etc.*), and $\boldsymbol{W}$ is the weight or load on the tyre.

By Newton's second law, the weight on the tyre depends on the fraction of the car's mass that the tyre must support and the acceleration of gravity, $\boldsymbol{g} = 32.1$ ft / sec^2. The fraction of the car's mass that the tyre must support depends on geometrical factors such as the wheelbase and the height of the centre of gravity. It also depends on the acceleration of the car, which completely accounts for weight transfer.

It is critical to separate the geometrical, or *kinematic*, aspects of weight transfer from the mass of the car. Imagine two cars with the same geometry but different masses (weights). In a one $\boldsymbol{g}$ braking manoeuvre, the same *fraction* of each car's total weight will be transferred to the front. In the example of Part 1 of this series, we calculated a 20% weight transfer during one *g* braking because the height of the CG was 20% of the wheelbase. This weight transfer will be the same 20% in a 3500 pound, stock Corvette as in a 2200 pound, tube-frame, Trans-Am Corvette so long as the geometry (wheelbase, CG height, *etc.*) of the two cars is the same. Although the actual weight, in pounds, will be different in the two cases, the fractions of the cars' total weight will be equal.

Separating kinematics from mass, then, we have for the weight $\boldsymbol{W = f(a)mg}$ where $\boldsymbol{f(a)}$ is the fraction of the car's mass the tyre must support and also accounts for weight transfer, $\boldsymbol{m}$ is the car's mass, and $\boldsymbol{g}$ is the acceleration of gravity.

Finally, by Newton's second law again, the acceleration of the tyre due to the force $\boldsymbol{F}$ applied to it is $\boldsymbol{a = F / f(a)m}$ We can now combine the expressions above to discover a fascinating fact:

$$\boldsymbol{a = F \big/ f(a)m \leq a_{max}}$$

$$\boldsymbol{a_{max} = \frac{\mu W}{f(a)m} = \frac{\mu f(a)mg}{f(a)m} = \mu g}$$

The maximum acceleration a tyre can take is $\boldsymbol{\mu g}$, a constant, independent of the mass of the car! While the maximum *force* a tyre can take depends very much on the current vertical load or weight on the tyre, the acceleration of that tyre does not depend on the current weight. If a tyre can take one $\boldsymbol{g}$ before sliding, it can take it on a lightweight car as well as on a heavy car, and it can take it under load as well as when lightly loaded. We hinted at this fact in Part 2, but the analysis above hopefully gives some deeper insight into it. We note that $\boldsymbol{a_{max}}$ being constant is only approximately true, because $\boldsymbol{\mu}$ changes slightly as tyre load varies, but this is a second-order effect (covered in a later article).

So, in an approximate way, we can consider the available acceleration from a tyre independently of details of weight transfer. The tyre will give you so many gees and that's that. This is the essential idea of the traction budget. What you do with your budget is your affair. If you have a tyre that will give you one *g*, you can use it for accelerating, braking, cornering, or some combination, but you cannot use more than your budget or you will slide. The front-back component of the budget measures accelerating and braking, and the right-left component measures cornering acceleration. The front-back component, call it $\boldsymbol{a}_y$, combines with the left-right component, $\boldsymbol{a}_x$, not by adding, but by the Pythagorean formula:

$$\boldsymbol{a} = \sqrt{\boldsymbol{a}_x^2 + \boldsymbol{a}_y^2}$$

Rather than trying to deal with this formula, there is a convenient, visual representation of the traction budget in the *circle of traction*. Figure 2 shows the circle. It is oriented in the same way as the X-ray view of the contact patch, Figure 1, so that up is forward and right is rightward. The circular boundary represents the limits of the traction budget, and every point inside the circle represents a particular choice of how you spend your budget. A point near the top of the circle represents pure, forward acceleration, a point near the bottom represents pure braking. A point near the right boundary, with no up or down component, represents pure rightward cornering acceleration. Other points represent Pythagorean combinations of cornering and forward or backward acceleration.

The beauty of this representation is that the effects of weight transfer are factored out. So the circle remains approximately the same no matter what the load on a tyre.

The Circle of Traction.

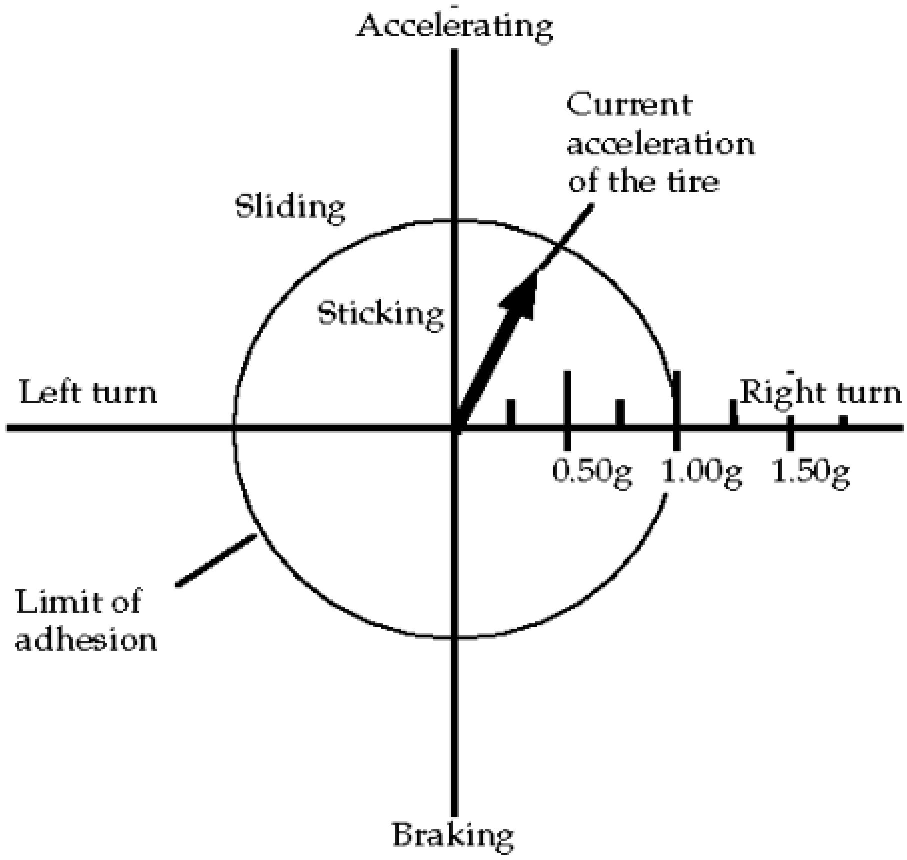


Figure 2: The Circle of Traction

In racing, of course, we try to spend our budget so as to stay as close to the limit, *i.e.*, the circular boundary, as possible. In street driving, we try to stay well inside the limit so that we have lots of traction available to react to unforeseen circumstances.

I have emphasized that the circle is only an approximate representation of the truth. It is probably close enough to make a computer driving simulation that feels right (I'm pretty sure that "Hard Drivin'" and other such games use it). As mentioned, tyre loads do cause slight, dynamic variations. Car characteristics also give rise to variations. Imagine a car with slippery tyres in the back and sticky tyres in the front. Such a car will tend to oversteer by sliding. Its traction budget will not look like a circle. Figure 3 gives an indication of what the traction budget for the whole car might look like (we have been discussing the budget of a single tyre up to this point, but the same notions apply to the whole car). In Figure 3, there is a large traction circle for the sticky front tyres and a small circle for the slippery rear tyres. Under acceleration, the slippery rears dominate the combined traction budget because of weight transfer. Under braking, the sticky fronts dominate. The combined traction budget looks something like an egg, flattened at top and wide in the middle. Under braking, the traction available for cornering is considerably greater than the traction available during acceleration because the sticky fronts are working. So, although this poorly handling car tends to oversteer by sliding the rear, it also tends to understeer during acceleration because the slippery rears will not follow the steering front tyres very effectively.

A traction budget diagram for a poorly handling car.

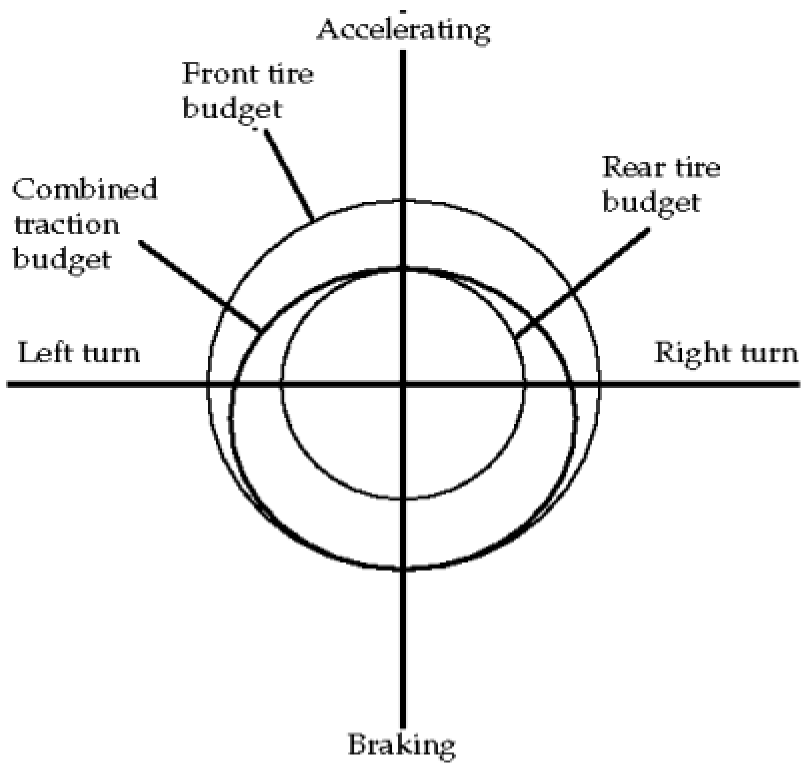


Figure 3: A traction budget diagram for a poorly handling car.

The traction budget is a versatile and simple technique for analysing and visualizing car handling. The same technique can be applied to developing driver's skills, planning the line around a course, and diagnosing handling problems.

The Physics of Racing, Part 8: Simulating Car Dynamics with a Computer Program

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

This month, we begin writing a computer program to simulate the physics of racing. Such a program is quite an ambitious one. A simple racing video game, such as “Pole Position,” probably took an expert programmer several months to write. A big, realistic game like “Hard Drivin’” probably took three to five people more than a year to create. The point is that the topic of writing a racing simulation is one that we will have to revisit many times in these articles, assuming your patience holds out. There are many “just physics” topics still to cover too, such as springs and dampers, transients, and thermodynamics. Your author hopes you will find the computer programming topic an enjoyable sideline and is interested, as always, in your feedback.

We will use a computer programming language called Scheme. You have probably encountered BASIC, a language that is very common on personal computers. Scheme is like BASIC in that it is *interactive*. An interactive computer language is the right kind to use when inventing a program as you go along. Scheme is better than BASIC, however, because it is a good deal simpler and also more powerful and modern. Scheme is available for most PCs at very modest cost (MIT Press has published a book and diskette with Scheme for IBM compatibles for about $40; I have a free version for Macintoshes). I will explain everything we need to know about Scheme as we go along. Although I assume little or no knowledge about computer programming on your part, we will ultimately learn some very advanced things.

The first thing we need to do is create a *data structure* that contains the mathematical state of the car at any time. This data structure is a block of computer memory. As simulated time progresses, mathematical operations performed on the data structure simulate the physics. We create a new instance of this data structure by typing the following on the computer keyboard at the Scheme prompt:

```
(new-race-car)
```

This is an example of an *expression*. The expression includes the parentheses. When it is typed in, it is *evaluated* immediately. When we say that Scheme is an interactive programming language, we mean that it evaluates expressions immediately. Later on, I show how we *define* this expression. It is by defining such expressions that we write our simulation program.

Everything in Scheme is an expression (that's why Scheme is simple). Every expression has a value. The value of the expression above is the new data structure itself. We need to give the new data structure a name so we can refer to it in later expressions:

```
(define car-161 (new-race-car))
```

This expression illustrates two Scheme features. The first is that expressions can contain sub-expressions inside them. The inside expressions are called *nested*. Scheme figures out which expressions are nested by counting parentheses. It is partly by nesting expressions that we build up the complexity needed to simulate racing. The second feature is the use of the special Scheme word `define`. This causes the immediately following word to become a stand-in synonym for the value just after. The technical name for such a stand-in synonym is *variable*. Thus, the expression `car-161`, wherever it appears after the `define` expression, is a synonym for the data structure created by the nested expression `(new-race-car)`.

We will have another data structure (with the same format) for `car-240`, another for `car-70`, and so on. We get to choose these names to be almost anything we like[1]. So, we would create all the data structures for the cars in our simulation with expressions like the following:

```
(define car-161 (new-race-car))
(define car-240 (new-race-car))
(define car-70  (new-race-car))
```

The state of a race car consists of several numbers describing the physics of the car. First, there is the car's position. Imagine a map of the course. Every position on the map is denoted by a pair of coordinates, ***x*** and ***y***. For elevation changes, we add a height coordinate, ***z***. The position of the centre of gravity of a car at any time is denoted with expressions such as the following:

```
(race-car-x car-161)
(race-car-y car-161)
(race-car-z car-161)
```

Each of these expressions performs *data retrieval* on the data structure `car-161`. The value of the first expression is the ***x*** coordinate of the car, *etc.* Normally, when running the Scheme interpreter, typing an expression simply causes its value to be printed, so we would see the car position coordinates printed out as we typed. We could also store these positions in another block of computer memory for further manipulations, or we could specify various mathematical operations to be performed on them.

The next pieces of state information are the three components of the car's velocity. When the car is going in any direction on the course, we can ask "how fast is it going in the ***x*** direction, ignoring its motion in the ***y*** and ***z*** directions?" Similarly, we want to know how fast it is going in the ***y*** direction, ignoring the ***x*** and ***z*** directions, and so on. Decomposing an object's velocity into separate components along the principal coordinate directions is necessary for computation. The technique was originated by the French mathematician Descartes, and Newton found that the motion in each direction can be analysed independently of the motions in the other directions at right angles to the first direction.

The velocity of our race car is retrieved via the following expressions:

```
(race-car-vx car-161)
(race-car-vy car-161)
(race-car-vz car-161)
```

To end this month's article, we show how velocity is computed. Suppose we retrieve the position of the car at simulated time t_1 and save it in some variables, as follows:

```
(define x1 (race-car-x car-161))
(define y1 (race-car-y car-161))
(define z1 (race-car-z car-161))
```

and again, at a slightly later instant of simulated time, t_2:

```
(define x2 (race-car-x car-161))
(define y2 (race-car-y car-161))
(define z2 (race-car-z car-161))
```

We have used define to create some new variables that now have the values of the car's positions at two times. To calculate the average velocity of the car between the two times and store it in some more variables, we evaluate the following expressions:

```
(define vx (/ (- x2 x1) (- t2 t1)))
(define vy (/ (- y2 y1) (- t2 t1)))
(define vz (/ (- z2 z1) (- t2 t1)))
```

The nesting of expressions is one level deeper than we have seen heretofore, but these expressions can be easily analysed. Since they all have the same form, it suffices to explain just one of them. First of all, the define operation works as before, just creating the variable vx and assigning it the value of the following expression. This expression is

```
(/ (- x2 x1) (- t2 t1))
```

In normal mathematical notation, this expression would read $\frac{x_2 - x_1}{t_2 - t_1}$ and in most computer languages, it would look like this:

```
(x2 - x1) / (t2 - t1)
```

We can immediately see this is the velocity in the x direction: a change in position divided by the corresponding change in time. The Scheme version of this expression looks a little strange, but there is a good reason for it: consistency. Scheme requires that all operations, including everyday mathematical ones, appear in the first position in a parenthesised expression, immediately after the left parenthesis. Although consistency makes mathematical expressions look strange, the payback is simplicity: all expressions have the same form. If Scheme had one notation for mathematical expressions and another notation for non-mathematical expressions, like most computer languages, it would be more complicated. Incidentally, Scheme's notation is called Polish notation. Perhaps you have been exposed to Hewlett-Packard calculators, which use reverse Polish, in which the operator always appears in the *last* position. Same idea, and advantages, as Scheme, only reversed.

So, to analyse the expression completely, it is a division expression

```
(/ ...)
```

whose two arguments are nested subtraction expressions

```
(- ...) (- ...)
```

The whole expression has the form

```
(/ (- ...) (- ...))
```

which, when the variables are filled in, is

```
(/ (- x2 x1) (- t2 t1))
```

After a little practice, Scheme's style for mathematics becomes second nature and the advantages of consistent notation pay off in the long run.

Finally, we should like to store the velocity values in our data structure. We do so as follows:

```
(set-race-car-vx! car-161 vx)
(set-race-car-vy! car-161 vy)
(set-race-car-vz! car-161 vz)
```

The set operations change the values in the data structure named **`car-161`**. The exclamation point at the end of the names of these operations doesn't do anything special. It's just a Scheme idiom for operations that change data structures.

Notes

1. It so happens, annoyingly, that we can't use the word **`car`**. This is a Scheme reserved word, like **`define`**. Its use is explained later.

The Physics of Racing, Part 9: Straights

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

We found in part 5 of this series, "Introduction to the Racing Line," that a driver can lose a shocking amount of time by taking a bad line in a corner. With a six-foot-wide car on a ten-foot-wide course, one can lose sixteen hundredths by "blowing" a single right-angle turn. This month, we extend the analysis of the racing line by following our example car down a straight. It is often said that the most critical corner in a course is the one before the longest straight. Let's find out how critical it is. We calculate how much time it takes to go down a straight as a function of the speed entering the straight. The results, which are given at the end, are not terribly dramatic, but we make several, key improvements in the mathematical model that is under continuing development in this series of articles. These improvements will be used as we proceed designing the computer program begun in Part 8.

The mathematical model for travelling down a straight follows from Newton's second law:

$$\boldsymbol{F} = \boldsymbol{ma} \qquad (1)$$

where $\boldsymbol{F}$ is the force on the car, $\boldsymbol{m}$ is the mass of the car, and $\boldsymbol{a}$ is the acceleration of the car. We want to solve this equation to get time as a function of distance down the straight. Basically, we want a table of numbers so that we can look up the time it takes to go any distance. We can build this table using accountants' columnar paper, or we can use the modern version of the columnar pad: the electronic spreadsheet program.

To solve equation (1), we first invert it:

$$\boldsymbol{a} = \boldsymbol{F} / \boldsymbol{m} \qquad (2)$$

Now $\boldsymbol{a}$, the acceleration, is the rate of change of velocity with time. *Rate of change* is simply the ratio of a small change in velocity to a small change in time. Let us assume that we have filled in a column of times on our table. The times start with 0 and go up by the same, small amount, say 0.05 sec. Physicists call this small time the *integration step*. It is standard practice to begin solving an equation with a fixed integration step. There are sometimes good reasons to vary the integration step, but those reasons do not arise in this problem. Let us call the integration step $\Delta \boldsymbol{t}$. If we call the time in the $\boldsymbol{i}$-th row $\boldsymbol{t_i}$, then for every row except the first,

$$\Delta t = t_i - t_{i-1} = \text{constant} \qquad (3)$$

We label another column *velocity*, and we'll call the velocity in the i-th row v_i. For every row except the first, equation (2) becomes:

$$\frac{v - v_{i-1}}{\Delta t} = F / m \qquad (4)$$

We want to fill in velocities as we go down the columns, so we need to solve equation (4) for v_i. This will give us a formula for computing v_i given v_{i-1} for every row except the first. In the first row, we put the speed with which we enter the straight, which is an input to the problem. We get:

$$v_i = v_{i-1} + \Delta t\, F / m \qquad (5)$$

We label another column *distance*, and we call the distance value in the i-th row x_i. Just as acceleration is the rate of change of velocity, so velocity is the rate of change of distance over time. Just as before, then, we may write:

$$v_i = \frac{x_i - x_{i-1}}{\Delta t} \qquad (6)$$

Solved for x_i, this is:

$$x_i = x_{i-1} + \Delta t v_i \qquad (7)$$

Equation (7) gives us a formula for calculating the distance for any time given the previous distance and the velocity calculated by equation (5). Physicists would say that we have a scheme for *integrating the equations of motion*.

A small detail is missing: what is the force, F? Everything to this point is *kinematic*. The real modelling starts now with formulas for calculating the force. For this, we will draw on all the previous articles in this series. Let's label another column *force*, and a few more with *drag*, *rolling resistance*, *engine torque*, *engine rpm*, *wheel rpm*, *trans gear ratio*, *drive ratio*, *wheel torque*, and *drive force*. As you can see, we are going to derive a fairly complete, if not accurate, model of accelerating down the straight. We need a few constants:

CONSTANT	SYMBOL	EXAMPLE VALUE
rear end ratio	R	3.07
density of air	ρ	0.0025 slugs / ft^3
coeff. of drag	C_d	0.30
frontal area	A	20 ft^2
wheel diameter	d	26 in = 2.167 ft

roll resist factor	r_r	0.696 lb / (ft / sec)
car mass	m	100 slug
first gear ratio	g_1	2.88
second gear ratio	g_2	1.91
third gear ratio	g_3	1.33
fourth gear ratio	g_4	1.00

and a few variables:

VARIABLE	SYMBOL	EXAMPLE
engine torque	T_E	330 ft-lbs
drag	F_d	45 lbs
rolling resistance	F_r	54 lbs
engine rpm	E	4000
wheel rpm	W	680
wheel torque	T_W	1930 ft-lbs
wheel force	F_W	1780 lbs
net force	F	1681 lbs

All the example values are for a late model Corvette. *Slugs* are the English unit of mass, and 1 slug weighs about 32.1 lbs at sea level (another manifestation of $F = ma$, with F in lbs, m in slugs, and a being the acceleration of gravity, 32.1 ft/sec^2).

The most basic modelling equation is that the force we can use for forward acceleration is the propelling force transmitted through the wheels minus drag and rolling resistance:

$$F = F_W - F_d - F_r \qquad (8)$$

The force of drag we get from Part 6:

$$F_d = \tfrac{1}{2} CdA \rho v_i^2 \qquad (9)$$

Note that to calculate the force at step i, we can use the velocity at step i. This force goes into calculating the acceleration at step i, which is used to calculate the velocity and distance at step $i + 1$ by equations (5) and (7). Those two equations represent the only "backward references" we need. Thus, the only inputs to the integration are the initial distance, 0, and the entrance velocity, v_0.

The rolling resistance is approximately proportional to the velocity:

$$F_r = r_r v_1 = 0.696 v_1 \qquad (10)$$

This approximation is probably the weakest one in the model. I derived it by noting from a Corvette book that 8.2 hp were needed to overcome rolling resistance at 55 mph. I have nothing else but intuition to go on for this equation, so take it with a grain of salt.

Finally, we must calculate the forward force delivered by the ground to the car by reaction to the rearward force delivered to the ground *via* the engine and drive train:

$$\boldsymbol{F}_w = \frac{\boldsymbol{T}_E \boldsymbol{R} \boldsymbol{g}_k}{\boldsymbol{d}/2} \qquad (11)$$

This equation simply states that we take the engine torque multiplied by the rear axle ratio and the transmission drive ratio in the ***k***-th gear, which is the torque at the drive wheels, $\boldsymbol{T}_W$, and divide it by the radius of the wheel, which is half the diameter of the wheel, ***d***.

To calculate the forward force, we must decide what gear to be in. The logic we use to do this is the following: from the velocity, we can calculate the wheel rpm:

$$\boldsymbol{W} = 60 \frac{\text{sec}}{\text{min}} \frac{\boldsymbol{v}_i}{\boldsymbol{\pi d}} \qquad (12)$$

From this, we know the engine rpm:

$$\boldsymbol{E} = \boldsymbol{W} \boldsymbol{R} \boldsymbol{g}_k \qquad (13)$$

At each step of integration, we look at the current engine rpm and ask "is it past the torque peak of the engine?" If so, we shift to the next highest gear, if possible. Somewhat arbitrarily, we assume that the torque peak is at 4200 rpm. To keep things simple, we also make the optimistic assumption that the engine puts out a constant torque of 330 ft-lbs. To make the model more realistic, we need merely look up a torque curve for our engine, usually expressed as a function of rpm, and read the torque off the curve at each step of the integration. The current approximation is not terrible however; it merely gives us artificially good times and speeds. Another important improvement on the logic would be to check whether the wheels are spinning, *i.e.*, that acceleration is less than about ½$\boldsymbol{g}$, and to "lift off the gas" in that case.

We have all the ingredients necessary to calculate how much time it takes to cover a straight given an initial speed. You can imagine doing the calculations outlined above by hand on columnar paper, or you can check my results (below) by programming them up in a spreadsheet program like Lotus 1-2-3 or Microsoft Excel. Eventually, of course, if you follow this series, you will see these equations again as we write our Scheme program for simulating car dynamics. Integrating the equations of motion by hand will take you many hours. Using a spreadsheet will take several hours, too, but many less than integrating by hand.

To illustrate the process, we show below the times and exit speeds for a 200 foot straight, which is a fairly long one in autocrossing, and a 500 foot straight, which you should only see on race tracks. We show times and speeds for a variety of speeds entering the straight from 25 to 50 mph in Table 1. The results are also summarized in the two plots, Figures (1) and (2).

Table 1: Exit speeds and times for several entrance speeds

	200 ft straight		**500 ft straight**	
Entrance speed (mph)	**Exit speed (mph)**	**Time (sec)**	**Exit speed (mph)**	**Time (sec)**
25	61.51	2.972	81.12	5.811
27	61.77	2.916	81.51	5.748
29	62.15	2.845	82.02	5.676
31	62.34	2.793	82.19	5.599
35	63.18	2.691	82.78	5.472
40	64.65	2.548	83.49	5.282
45	66.85	2.392	84.68	5.065
50	69.27	2.261	85.83	4.875

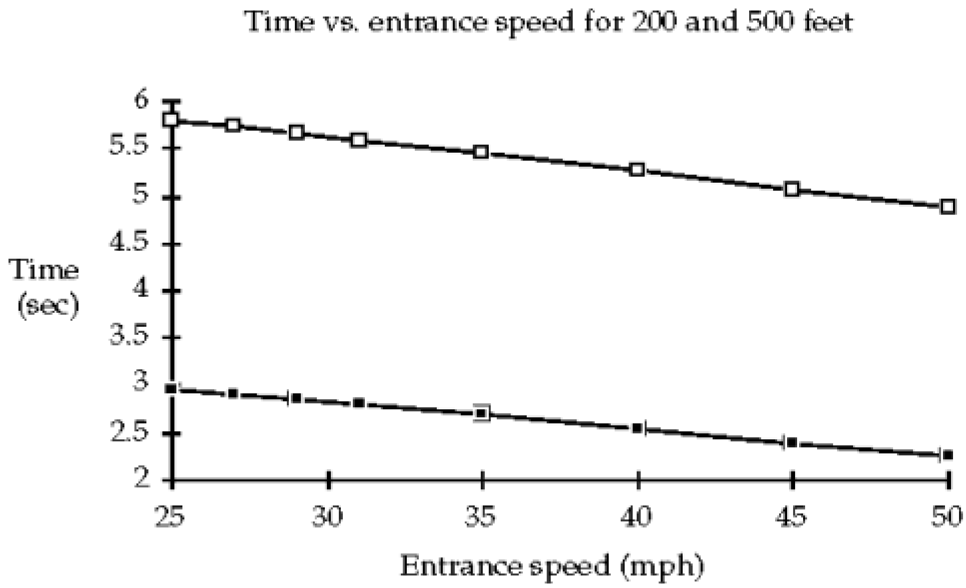


Figure 1:

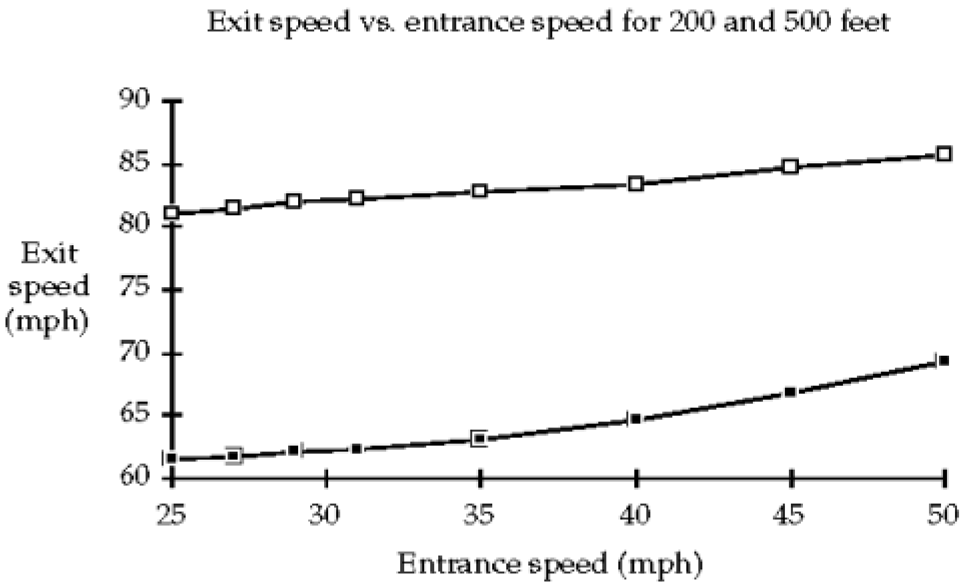


Figure 2:

The notable facts arising in this analysis are the following. The time difference resulting from entering the 200' straight at 27 mph rather than 25 mph is about 6 hundredths. Frankly, not as much as I expected. The time difference between entering at 31 mph over 25 mph is about 2 tenths, again less than I would have guessed. The speed difference at the end of the straight between entering at 25 mph and 50 mph is only 8 mph, a result of the fact that the car labours against friction and higher gear ratios at high speeds. It is also a consequence of the fact that there is so much torque available at 25 mph in low gear that the car can almost make up the difference over the relatively short 200' straight. In fact, on the longer 500' straight, the exit speed difference between entering at 25 mph and 50 mph is not even 5 mph, though the time difference is nearly a full second.

This analysis would most likely be much more dramatic for a car with less torque than a Corvette. In a Corvette, with 330 ft-lbs of torque on tap, the penalty for entering a straight slower than necessary is not so great as it would be in a more typical car, where recovering speed lost through timidity or bad cornering is much more difficult.

Again, the analysis can be improved by using a real torque curve and by checking whether the wheels are spinning in lower gears.

The Physics of Racing, Part 10: Grip Angle

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

In many ways, tyre mechanics is an unpleasant topic. It is shrouded in uncertainty, controversy, and trade secrecy. Both theoretical and experimental studies are extremely difficult and expensive. It is probably the most uncontrollable variable in racing today. As such, it is the source of many highs and lows. An improvement in modelling or design, even if it is found by lucky accident, can lead to several years of domination by one tyre company, as with BFGoodrich in autocrossing now. An unfortunate choice of tyre by a competitor can lead to frustration and a disastrous hole in the budget.

This month, we investigate the physics of tyre adhesion a little more deeply than in the past. In Parts 2, 4, and 7, we used the simple friction model given by $\boldsymbol{F} \leq \boldsymbol{\mu W}$, where $\boldsymbol{F}$ is the maximum traction force available from a tyre; $\boldsymbol{\mu}$, assumed constant, is the coefficient of friction; and $\boldsymbol{W}$ is the instantaneous vertical load, or weight, on a tyre. While this model is adequate for a rough, intuitive feel for tyre behaviour, it is grossly inadequate for quantitative use, say, for the computer program we began in Part 8 or for race car engineering and set up.

I am not a tyre engineer. As always, I try to give a fresh look at any topic from a physicist's point of view. I may write things that are heretical or even wrong, especially on such a difficult topic as tyre mechanics. I invite debate and corrections from those more knowledgeable than I. Such interaction is part of the fun of these articles for me.

I call this month's topic "grip angle." The grip angle is a quantity that captures, for many purposes, the complex and subtle mechanics of a tyre. Most writers call this quantity "slip angle." I think this name is misleading because it suggests that a tyre works by slipping and sliding. The truth is more complicated. Near maximum loads, the contact patch is partly gripping and partly slipping. The maximum net force a tyre can yield occurs at the threshold where the tyre is still gripping but is just about to give way to total slipping. Also, I have some difficulties with the analyses of slip angle in the literature. I will present these difficulties in these articles, unfortunately, probably without resolution. For these reasons, I give the quantity a new name.

A tyre is an elastic or deformable body. It delivers forces to the car by stretching, compressing, and twisting. It is thus a very complex sort of spring with several different ways, or *modes*, of deformation. The hypothetical tyre implied by

$\boldsymbol{F} \leq \boldsymbol{\mu W}$ with constant $\boldsymbol{\mu}$ would be a non-elastic tyre. Anyone who has driven hard tyres on ice knows that non-elastic tyres are basically uncontrollable, not just because $\boldsymbol{\mu}$ is small but because regular tyres on ice do not twist appreciably.

The first and most obvious mode of deformation is radial. This deformation is along the radius of the tyre, the line from the centre to the tread. It is easily visible as a bulge in the sidewall near the contact patch, where the tyre touches the ground. Thus, radial compression varies around the circumference.

Second is circumferential deformation. This is most easily visible as wrinkling of the sidewalls of drag tyres. These tyres are intentionally set up to deform dramatically in the circumferential direction.

Third is axial deformation. This is a deflection that tends to pull the tyre off the (non-elastic) wheel or rim.

Last, and most important for cornering, is *torsional* deformation. This is a difference in axial deflection from the front to the back of the contact patch. Fundamentally, radial, circumferential, and axial deformation furnish a complete description of a tyre. But it is very useful to consider the differences in these deflections around the circumference.

Let us examine exactly how a tyre delivers cornering force to the car. We can get a good intuition into the physics with a pencil eraser. Get a block eraser, of the rectangular kind like “Pink Pearl” or “Magic Rub.” Stand it up on a table or desk and think of it as a little segment of the circumference of a tyre. Think of the part touching the desk as the contact patch. Grab the top of the eraser and think of your hand as the wheel or rim, which is going to push, pull, and twist on the segment of tyre circumference as we go along the following analysis.

Consider a car travelling at speed $\boldsymbol{v}$ in a straight line. Let us turn the steering wheel slightly to the right (twist the top of the eraser to the right). At the instant we begin turning, the rim (your hand on the eraser), at a circumferential position just behind the contact patch, pushes slightly leftward on the bead of the tyre. Just ahead of the contact patch, likewise, the rim pulls the bead a little to the right. The push and pull together are called a *force couple*. This couple delivers a torsional, clockwise stress to the inner part of the tyre carcass, near the bead. This stress is communicated to the contact patch by the elastic material in the sidewalls (or the main body of the eraser). As a result of turning the steering wheel, therefore, the rim twists the contact patch clockwise.

The car is still going straight, just for an instant. How are we going to explain a net rightward force from the road on the contact patch? This net force must be there, otherwise the tyre and the car would continue in a straight line by Newton’s First Law.

Consider the piece of road just under the contact patch at the instant the turn begins. The rubber particles on the left side of the patch are going a little bit faster with respect to the road than the rest of the car and the rubber particles on the right side of the patch are going a little bit slower than the rest of the car. As a result, the left side of the patch grips a little bit less than the right. The rubber particles on the left are more likely to slide and the ones on the right are more likely to grip. Thus, the left edge of the patch "walks" a little bit upward, resulting in a net clockwise twisting motion of the patch. The torsional stress becomes a torsional motion. As this motion is repeated from one instant to the next, the tyre (and the eraser-I hope you are still following along with the eraser) walks continuously to the right.

The better grip on the right hand side of the contact patch adds up to a net rightward force on the tyre, which is transmitted back through the sidewall to the car. The chassis of the car begins to yaw to the right, changing the direction of the rear wheels. A torsional stress on the rear contact patches results, and the rear tyres commence a similar "walking" motion.

The wheel (your hand) is twisted more away from the direction of the car than is the contact patch. The angular difference between the direction the wheel is pointed and the direction the tyre walks is the grip angle. All quantities of interest in tyre mechanics-forces, friction coefficients, *etc.*, are conventionally expressed as functions of grip angle.

In steady state cornering, as in sweepers, an understeering car has larger grip angles in front, and an oversteering car has larger grip angles in the rear. How to control grip angles statically with wheel alignment and dynamically with four-wheel steering are subjects for later treatment.

The greater the grip angle, the larger the cornering force becomes, up to a point. After this point, greater grip angle delivers less force. This point is analogous to the idealized adhesive limit mentioned earlier in this series. Thus, a real tyre behaves qualitatively like an ideal tyre, which grips until the adhesive limit is exceeded and then slides. A real tyre, however, grips gradually better as cornering force increases, and then grips gradually worse as the limit is exceeded.

The walking motion of the contact patch is not entirely smooth, or in other words, somewhat *discrete*. Individual blocks of rubber alternately grip and slide at high frequency, thousands of times per second. Under hard cornering, the rubber blocks vibrating on the road make an audible squalling sound. Beyond the adhesive limit, squealing becomes a lower frequency sound, "squalling," as the point of optimum efficiency of the walking process is bypassed.

There is a lot more to say on this subject, and I admit that my first attempts at a mathematical analysis of grip angle and contact patch mechanics got bogged down. However, I think we now have an intuitive, conceptual basis for better modelling in the future.

Speaking of the future, summarizing briefly the past of and plans for the *Physics of Racing* series. The following overlapping threads run through it:

Tyre Physics

concerns adhesion, grip angle, and elastic modelling. This has been covered in Parts 2, 4, 7, and 10, and will be covered in several later parts.

Car Dynamics

concerns handling, suspension movement, and motion of a car around a course; has been covered in Parts 1, 4, 5, and 8 and will continue.

Drive Line Physics

concerns modelling of engine performance and acceleration. Has been covered in Parts 3, 6, and 9 and will also continue.

Computer Simulation

concerns the design of a working program that captures all the physics. This is the ultimate goal of the series. It was begun in Part 8 and will eventually dominate discussion.

The following is a list of articles that have appeared so far:

1. Weight Transfer
2. Keeping Your Tyres Stuck to the Ground
3. Basic Calculations
4. There is No Such Thing as Centrifugal Force
5. Introduction to the Racing Line
6. Speed and Horsepower
7. The Circle of Traction
8. Simulating Car Dynamics with a Computer Program
9. Straights
10. Grip Angle

and the following is a tentative list of articles I have planned for the near future (naturally, this list is "subject to change without notice"):

Springs and Dampers,

presenting a detailed model of suspension movement (suggested by Bob Mosso)

Transients,

presenting the dynamics of entering and leaving corners, chicanes, and slaloms (this one suggested by Karen Babb)

Stability,

explaining why spins and other losses of control occur

Smoothness,

exploring what, exactly, is meant by smoothness

Modelling Car Data

in a computer program; in several articles

Modelling Course Data

in a computer program; also in several articles

In practice, I try to keep the lengths of articles about the same, so if a topic is getting too long (and grip angle definitely did), I break it up in to several articles.

The Physics of Racing, Part 11: Braking

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

I was recently helping to crew Mark Thornton's effort at the Silver State Grand Prix in Nevada. Mark had built a beautiful car with a theoretical top speed of over 200 miles per hour for the 92 mile time trial from Lund to Hiko. Mark had no experience driving at these speeds and asked me as a physicist if I could predict what braking at 200 mph would be like. This month I report on the back-of-the-envelope calculations on braking I did there in the field.

There are a couple of ways of looking at this problem. Brakes work by converting the energy of motion, kinetic energy, into the energy of heat in the brakes. Converting energy from useful forms (motion, electrical, chemical, *etc.*) to heat is generally called dissipating the energy, because there is no easy way to get it back from heat. If we assume that brakes dissipate energy at a constant rate, then we can immediately conclude that it takes four times as much time to stop from 200 mph as from 100 mph. The reason is that kinetic energy goes up as the square of the speed. Going at twice the speed means you have four times the kinetic energy because $4 = 2^2$. The exact formula for kinetic energy is $\frac{1}{2}\boldsymbol{mv}^2$, where $\boldsymbol{m}$ is the mass of an object and $\boldsymbol{v}$ is its speed. This was useful to Mark because braking from 100 mph was within the range of familiar driving experience.

That's pretty simple, but is it right? Do brakes dissipate energy at a constant rate? My guess as a physicist is "probably not." The efficiency of the braking process, dissipation, will depend on details of the friction interaction between the brake pads and disks. That interaction is likely to vary with temperature. Most brake pads are formulated to grip harder when hot, but only up to a point. Brake fade occurs when the pads and rotors are overheated. If you continue braking, heating the system even more, the brake fluid will eventually boil and there will be no braking at all. Brake fluid has the function of transmitting the pressure of your foot on the pedal to the brake pads by hydrostatics. If the fluid boils, then the pressure of your foot on the pedal goes into crushing little bubbles of gaseous brake fluid in the brake lines rather than into crushing the pads against the disks. Hence, no brakes.

We now arrive at the second way of looking at this problem. Let us assume that we have good brakes, so that the braking process is limited not by the interaction between the pads and disks but by the interaction between the tyres and the ground. In other words, let us assume that our brakes are better than our tyres. To keep things simple and back-of-the-envelope, assume that our tyres will give us a constant deceleration of

$$1\boldsymbol{g} \equiv \boldsymbol{a} = 32.1\frac{\text{feet}}{\text{sec}^2}$$

The time $\boldsymbol{t}$ required for braking from speed $\boldsymbol{v}$ can be calculated from: $\boldsymbol{t} = \boldsymbol{v} / \boldsymbol{a}$ which simply follows from the definition of constant acceleration. Given the time for braking, we can calculate the distance $\boldsymbol{x}$, again from the definitions of acceleration and velocity:

$$\boldsymbol{x} = \boldsymbol{vt} - \tfrac{1}{2}\boldsymbol{at}^2$$

Remembering to be careful about converting miles per hour to feet per second, we arrive at the numbers in Table 1.

Starting Speed (mph)	Starting Speed (fps)	Time to brake (sec)	Distance to brake (feet)	Distance to brake (yards)
30	44	1.37	30.16	10.05
60	88	2.74	120.62	40.21
90	132	4.11	271.40	90.47
120	176	5.48	482.49	160.83
150	220	6.85	753.89	251.30
180	264	8.22	1085.61	361.87
210	308	9.60	1477.63	492.54

Table 1: Times and Distances for barking to zero from various speeds

We can immediately see from this table (and, indeed, from the formulas) that it is the *distance*, not the time, that varies as the square of the starting speed $\boldsymbol{v}$. The braking time only goes up linearly with speed, that is, in simple proportion.

The numbers in the table are in the ballpark of the braking figures one reads in published tests of high performance cars, so I am inclined to believe that the second way of looking at the problem is the right way. In other words, the assumption that the brakes are better than the tyres, so long as they are not overheated, is probably right, and the assumption that brakes dissipate energy at a constant rate is probably wrong because it leads to the conclusion that braking takes more time than it actually does.

My final advice to Mark was to leave *lots of room*. You can see from the table that stopping from 210 mph takes well over a quarter mile of very hard, precise, threshold braking at 1$\boldsymbol{g}$!

The Physics of Racing, Part 12: CyberCar, Every Racer's DWIM Car?

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

The cybernetic DWIM car is coming. DWIM stands for "Do What I Mean[1]." It is a commonplace term in the field of Human-machine Interfaces, and refers to systems that automatically interpret the user's intent from his or her inputs.

Cybernetics (or at least one aspect of it) is the science of unifying humans and machines. The objective of cybernetics is usually to amplify human capability with "intelligent" machines, but sometimes the objective is the reverse. Most of the work in cybernetics has been under the aegis of defence, for building advanced tanks and aircraft. There is a modest amount of cybernetics in the automotive industry, as well. Anti-lock Braking (ABS), Acceleration Slip Reduction (ASR), Electronic Engine Management, and Automatic Traction Control (ATC) are cybernetic DWIM systems – of a kind – already in production. They all make "corrections" on the driver's input based on an assumed intention. Steer-by-wire, Continuously Variable Transmissions (CVT), and active suspensions are on the immediate horizon. All these features are part of a distinct trend to automate the driving experience. This month, we take a break from hard physics to look at the better and the worse of increased automation, and we look at one concept of the ultimate result, CyberCar.

Among the research directions in cybernetics are advanced sensors for human inputs. One of the more incredible is a system that reads brain waves and figures out what a fighter pilot wants to do directly from patterns in the waves.

A major challenge in the fighter cockpit is information overload. Pilots have far too many instruments, displays, horns, buzzers, radio channels, and idiot lights competing for their attention. In stressful situations, such as high speed dogfights, the pilot's brain simply ignores inputs beyond its capacity, so the pilot may not hear a critical buzzer or see a critical warning light. In the "intelligent cockpit," however, the pilot *consciously* suppresses certain displays and auditory channels, thus reducing sensory clutter. By the same token, the intelligent cockpit must be able to override the pilot's choices and to put up critical displays and to sound alarms in emergencies. In the reduced clutter of the cockpit, then, it is much less likely that a pilot will miss critical information.

How does the pilot select the displays that he[2] wants to see? The pilot cannot afford the time to scroll through menus like those on a personal computer screen or hunt-and-peck on a button panel like that on an automatic bank teller machine.

There are already sensors that can read a pilot's brain waves and anticipate what he wants to look at next. Before the pilot even consciously knows that he wants to look at a weapon status display, for example, the cybernetic system can infer the intention from his brain waves and pop up the display. If he thinks it is time to look at the radar, before he could speak the command, the system reads his brain waves, pops up the radar display, and puts away the weapon status display.

How does it work? During a training phase, the system reads brain waves and gets explicit commands through a button panel. The system analyses the brain waves, looking for certain unique features that it can associate with the intention (inferred from the command from the button panel) to see the radar display, and other unique features to associate with the intention to look at weapon status, and so on. The system must be trained individually for each pilot. Later, during operation, whenever the system sees the unique brain wave patterns, it "knows" what the pilot wants to do.

The implications of technology like this for automobiles is amazing. Already, things like ABS are a kind of rudimentary cybernetics. When a driver stands all over the brake pedal, it is assumed that his intention is to stop, not to skid. The ABS system "knows," in a manner of speaking, the driver's intention and manages the physical system of the car to accomplish that goal. So, instead of being a mere mechanical linkage between your foot and the brakes, the brake pedal becomes a kind of intentional, DWIM control. Same goes for traction control and ASR. When the driver is on the gas, the system "knows" that he wants to go forward, not to spin out or do doughnuts. In the case of TC, the system regulates the torque split to the drive wheels, whether there be two or four. In the case of ASR, the system backs off the throttle when there is wheel spin. Cybernetics again.

ABS, TC, and ASR exist now. What about the future? Consider steer-by-wire. CyberCar, the total cybernetic car, infers the driver's intended direction from the steering wheel position. It makes corrections to the actual direction of the steered wheels and to the throttle and brakes much more quickly and smoothly than any driver can do. Coupled with slip angle[3] sensors[1] and inertial guidance systems, perhaps based on miniaturized laser/fibre optic gyros (no moving parts), cybernetic steering, throttle, and brake controls will make up a formidable racing car that could drive a course in practically optimal fashion given only the driver's *desired* racing line.

In an understeering situation, when a car is not turning as much as desired, a common driver mistake is to turn the steering wheel more. That is a mistake, however, only because the driver is treating the steering wheel as an *intentional* control rather than the physical control it actually is. In CyberCar, however, the steering wheel *is* an intentional control. When the driver adds more lock in a corner, CyberCar "knows" that the driver just wants more steering. Near the limits of adhesion, CyberCar knows that the appropriate *physical* reaction is, in fact, some weight transfer to the front, either by trailing throttle or a little braking, and a little less steering wheel lock. When the fronts hook up again, CyberCar can immediately get back into the throttle and add a little more steering lock, all the while tracking the driver's desires through the intentional steering wheel in the cockpit. Similarly, in an oversteer situation, when the driver gives opposite steering lock, CyberCar knows what to do. First, CyberCar determines whether the condition is trailing throttle oversteer (TTO) or power oversteer (PO). It can do this by monitoring tyre loads through suspension deflection and engine torque output over time. In TTO, CyberCar adds a little throttle and counter steers. When the drive wheels hook up again, it modulates the throttle and dials in a little forward lock. In PO, CyberCar gently trails off the throttle and counter steers. All the while, CyberCar monitors driver's intentional inputs and the physical status of the car at the rate of several kilohertz (thousands of times per second).

The very terms "understeer" and "oversteer" carry cybernetic implication, for these are terms of intent. Understeer means the car is not steering as much as wanted, and oversteer means it is steering too much.

The above description is within current technology. What if we get *really* fantastic? How about doing away with the steering wheel altogether? CyberCar, version II, knows where the driver wants to go by watching his eyes, and it knows whether to accelerate or brake by watching brain waves. With Virtual Reality and teleoperation, the driver does not even have to be inside the car. The driver, wearing binocular video displays that control in-car cameras (or even synthetic computer graphics) *via* head position, sits in a virtual cockpit in the pits.

Now we must ask how much cybernetics is desirable? Autocrossing is, largely, a pure driver skill contest. Wheel-to-wheel racing adds race craft – drafting, passing, deception, *etc.* – to car control skills. Does it not seem that cybernetics eliminates driver skill as a factor by automating it? Is it not just another way for the "haves" to beat the "have-nots" by out-spending them? Drivers who do not have ABS have already complained that it gives their competition an unfair advantage. On the other hand, drivers who *do* have it have complained that it reduces their feel of control and their options while braking. I think they doth protest too much.

In the highest forms of racing, where money is literally no object, cybernetics is already playing a critical role. The clutch-less seven speed transmissions of the Williams/Renault team dominated the latter half of the 1991 Formula 1 season. But for some unattributable bad luck, they would have won the driver's championship and the constructor's cup. Carrol Smith, noted racing engineer, has been predicting for years that ABS will show up in Formula 1 as soon as systems can be made small and light enough[2]. It seems inevitable to me that cybernetic systems will give the unfair advantage to those teams most awash in money. However, autocrossers, club racers, and other grass roots competitors will be spared the expense, and the experience of being relieved of the enjoyment of car control, for at least another decade or two.

Acknowledgements

Thanks to Phil Ethier for giving me a few tips on car control that I might be able to teach to CyberCar and to Ginger Clark for bringing slip angle sensors to my attention.

Notes

1 And the word play on 'dream' was too much to resist.
2 Everywhere, 'he' means 'he or she,' 'his' means 'his or her,' *etc.*
3 Also known as *grip angle*; see Part 10 of this series.

References

[1] Patrick Borthelow, "Sensing Tire Slip Angles At the Racetrack," *Sensors*, September 1991.

[2] Carrol Smith, *Engineer to Win*, *Prepare to Win*, *Build to Win*, from Classic Motorbooks, P.O. Box 1/RT021, Osceola, WI, 54020.

The Physics of Racing, Part 13: Transients

Brian Beckman

physicist and member of
No Bucks Racing Club

P.O. Box 662
Burbank, CA 91503

Obviously, handling is extremely important in any racing car. In an autocross car, it is critical. A poorly handling car with lots of power will not do well at all on the typical autocross course. A Miata or CRX can usually beat a 60's muscle car like a Pontiac GTO even though the Goat may have four or five times the power. Those cars, while magnificently powerful, were designed for straight-line acceleration at the expense of cornering.

This month, we examine one aspect of handling, that of handling *transient* or short-lived forces. Usually, in motor sports contexts, the word "transient" means short-lived cornering forces as opposed to braking and accelerating forces. In broader contexts, it means any short-lived forces.

Transients figure prominently in autocross. Perhaps the epitome of a transient-producing autocross feature is slalom, which requires a car and driver to flick quickly from left to right and back again. Many courses also feature esses, lane changes, chicanes (dual lane changes), alternating gates, and other variations on the theme. All of these require quick cornering response to transients. Some sports cars, like Elans, MR2's and X1/9's, are designed specifically to have such quick response. The general rule is that these kinds of cars get you into a corner more quickly than do other kinds. They achieve their response with low weight and low *polar moment of inertia* (PMI). A chief goal of this article is to explain PMI.

Most engineering designs are trade-offs, and designing for quick transient response is no exception. Light weight means, generally, a small engine. Low PMI means, generally, placing the engine as close as possible to the centre of mass (CM) as possible. So, many quick response cars are mid-engined, further constraining engine size. With engine size, we get into another trade-off area: cost versus power. Smaller engines are, generally, less powerful. The cheapest way to get engine power is with size. A big, sloppy, over-the-counter American V8 can cheaply give you 300-400 ft-lb of torque. Getting the same torque from a 1.6 litre four-banger can be very expensive and will put you firmly in the Prepared or Modified ranks. But, a bigger engine is a heavier engine, and that means a beefier (heavier) frame and suspension to support it. Therefore, the cheap way to high torque requires sacrificing some transient response for power. This design approach is typified by Corvettes and Camaros. The general rule is that these kinds of cars get you out of a corner more quickly because of engine torque.

So, we can divide the sports car universe into the lightweight, quick-response-style camp and the ground-thumping, stump-pulling-style camp. Some cars straddle the boundary and try to be lightweight, with low PMI, and powerful. These cars are usually very expensive because the fundamental design compromises are pushed with exotic materials and great amounts of engineer time. Ordinary cars are usually mostly one or the other. No one can say which style is “better.” Both kinds of car are great fun to drive. There are some courses on which quick-response type cars will have top times and others on which the V8’s will be unbeatable. Fortunately, these two styles of cars are usually in different classes.

Let’s back up that discussion with some physics. What is transient response and how does it relate to polar moment of inertia?

Any object resists a change in its state of motion. If it is not moving, it resists moving. If it is moving, it resists stopping or changing direction. The resistance is generally called *inertia*. With straight line motion, inertia has only one aspect: *mass*. Handling is mostly about cornering, however, not about straight-line motion.

Cornering is a change in the direction of motion of a car. In order to change the direction of motion, we must change the direction in which the car is pointing. To do that, we must rotate or *yaw* the car. However, the car will resist yawing because the various parts of the car will resist changing their states of motion. Let’s say we are cornering to the right, hence yawing clockwise. The suspension parts and frame and cables and engine *etc. etc.* in the front part of the car will resist veering to the right off their prior straight-line course and the suspension parts and frame and differential and gas tank *etc. etc.* in the rear will resist veering to the left off their prior straight-line course. From this observation, we can ‘package’ the inertial resistance to yawing of any car into a convenient quantity, the PMI. What follows is a simplified, two dimensional analysis. The full, three-dimensional case is conceptually similar though more complicated mathematically.

It turns out that the general motion of any large object can be broken up into the motion of the centre of mass, treated as a small particle, and the rotation of the object about its centre of mass. This means that to do dynamical calculations that account for cornering, we must apply Newton’s Second Law, $\boldsymbol{F} = \boldsymbol{ma}$, *twice*. First, we apply the law to all masses in the car taken as an aggregate with their positions measured with respect to a fixed point on the ground. Second, we apply the law individually to the massive parts of the car with their positions measured from the CM in the car while it moves.

Let’s make a list of all the N parts in the car. Let the variable i run over all the limits in the list; let the masses of the parts m_i, their positions on the X axis of the ground coordinate grid be x_i and their positions on the Y axis of the ground coordinate grid be y_i. We summarise the position information with *vector* notation, writing a bold character, $\boldsymbol{r_i}$, for the position of the i-th part. Vector notation saves us from having to write two (or three) sets of equations, one for each coordinate direction on the grid. For many purposes, a vector can be treated like a number in symbolic arithmetic. We must break a vector equation apart into its constituent *component* equations when it’s time to do number-crunching.

The (vector) position $\boldsymbol{R}$ of the CM with respect to the ground is just the mass weighted average over all the parts of the car:

$$\boldsymbol{R} = \frac{\sum_{i-1}^{N} m_i \boldsymbol{r}_i}{M = \sum_{i-1}^{N} m_i} \tag{1}$$

The external forces on the car are also vectors: they have X components and Y components. So, we write the sum of all the forces on the car with a bold $\boldsymbol{F}$. Similarly, the velocity of the CM is a vector. It is the change in $\boldsymbol{R}$ over a small time, dt, divided by the time. This is written

$$\boldsymbol{V} = \frac{d\boldsymbol{R}}{dt} \tag{2}$$

The d/dt notation is called a *derivative*. In turn, the acceleration is a small change in the velocity divided by the time:

$$\boldsymbol{A} = \frac{d\boldsymbol{V}}{dt} = \frac{d^2\boldsymbol{R}}{dt^2} \tag{3}$$

The d^2/dt^2 notation is called a *second derivative* and results from two derivatives in succession.

Newton's Second Law for the CM of the car is then

$$\boldsymbol{F} = M\frac{d^2\boldsymbol{R}}{dt^2} \tag{4}$$

where M is the total mass of all parts in the car. Simple, eh? This is a *differential equation*, and theoretical physics is overwhelmingly concerned with the solutions of such things. In this case, a solution is finding $\boldsymbol{R}$ given M and $\boldsymbol{F}$. We can also simplify the writing of the equations in general by replacing time-derivative notations with dots: one dot for one time derivative and two dots for two derivatives. We get

$$\boldsymbol{F} = M\ddot{\boldsymbol{R}} \tag{5}$$

Now, we consider the parts of the car separately as they yaw (and pitch and roll) about the CM while remaining firmly attached to the car. Let's write all position variables measured with respect to the coordinate grid fixed in the car with overbars, so the vector position of the i-th mass in our list is r_i.

However, we don't need to use vectors (in two dimensions), because in pure yawing motion about the CM of the car, the radial distance of each car part from the CM remains fixed and each part has the same yaw angle as the whole car.

Let the yaw angle of the car and its coordinate grid measured against the ground-based, inertial coordinates be θ. As each part is affected by forces, it moves in a yaw-arc around the CM. A small amount of yaw is written $d\theta$. Each part moves perpendicularly to a line drawn from the part to the CM of the car, and the distance it moves is equal to its radial distance from the CM, r_i (non-bold: a number, not a vector), times the little amount of yaw $d\theta$. Divide by the little time over which the motions are measured, and you have the velocity of each car part:

$$\bar{v}_i = \bar{r}_i \frac{d\theta}{dt} = \bar{r}_i \theta \qquad (6)$$

Now, it's easy to apply Newton's second law. Equate the force on the i-th part F_i, to the mass of the part times the acceleration of the part:

$$\bar{F}_i = m_i \bar{r}_i \ddot{\theta} \qquad (7)$$

We're almost done with the math, so hang in there. If we multiply equation (7) by r_i on both sides, the left-hand side becomes the torque of the forces on the i-th part about the CM:

$$\bar{\Lambda}_i = \bar{r}_i \bar{F}_i = m_i \bar{r}_i^{\,2} \ddot{\theta} \qquad (8)$$

Now, if we sum this equation up over all the parts in our list, we can drop the i subscript:

$$\bar{\Lambda}_i = \left(\sum_{i=1}^{N} m_i \bar{r}_i^{\,2} \right) \ddot{\theta} \qquad (9)$$

remembering that all parts have the same $\ddot{\theta}$. The reason for doing this is that resulting equation *looks like* Newton's Second Law, equation (5). If you replace $\sum m_i \bar{r}_i^2$ with a symbol, I, the equation is identical in form:

$$\bar{\Lambda} = \bar{I} \ddot{\theta} \qquad (10)$$

Physicists like to find formal equivalences amongst equations because they can use the same mathematical techniques to solve all of them. The equivalences also hints at deeper insights into similarities in the Universe.

OK, if you haven't already guessed it, $\bar{I} = \sum m_i \bar{r}_i^2$ is the polar moment of inertia. To compute it for a given car, we take all the parts in the car, measure their masses and their distances from the CM, square, multiply and add. In practice, this is very difficult. I doubt if PIMs are measured very often, but when they are, it is probably done experimentally: by subjecting the car to known torques and measuring how quickly yaw angle accumulates.

We can also see that, for a given rotational torque, the acceleration of yaw angle is inversely proportional to I. Thus, we have backed up, from first principles, our statement that cars with low PMI respond more quickly, by yawing, to transient cornering forces than do cars with large PMI. A car with a low PMI is designed so that the heavy parts – primarily the engine – are as close to the CM as possible. Moving the engine even a couple of inches closer to the CM can dramatically decrease the PMI because it varies as the *square* of the distance of parts from the CM.

Since equation (10) is formally equivalent to Newton's Second Law, an analogous insight applies to that law. A car with low mass responds more quickly to forces with straight-line changes in motion just as a car with low PMI responds more quickly to torques with rotational changes in motion.

Why would one design a car with a high PMI? Only to get a big, powerful engine into it that might have to be placed in the front or the rear, far from the CM. So, take your pick. Choose a car with a low PMI that yaws very quickly and give up on some engine power. Or, choose a car with colossal engine and give up on some handling quickness.

The Physics of Racing, Part 14: Why Smoothness?

Brian Beckman PhD

I'm back after a hiatus of nine years. Time does fly, doesn't it? For those counting articles, the last one published was part 12; there is no Part 13.

After such a long time away, it might be worthwhile to repeat the motivation and goals of this "Physics of Racing" series. I am a physicist (the "PhD" after my name is from my Union card). I'm also an active participant in motorsports. It would be almost impossible for me not to use my professional training to analyse my hobby. So, I've been thinking for some time about the physics of racing cars.

Part of the fun for me is to do *totally original* analyses. As such, they won't have the specifics of a hardcore engineering analysis. You can look that up in books by Fred Puhn, William Milliken, and Carrol Smith, amongst many others. I want to find the bare-bones physics behind the engineering – at the risk of bypassing some detail. In sum, I analyse things completely from scratch because:

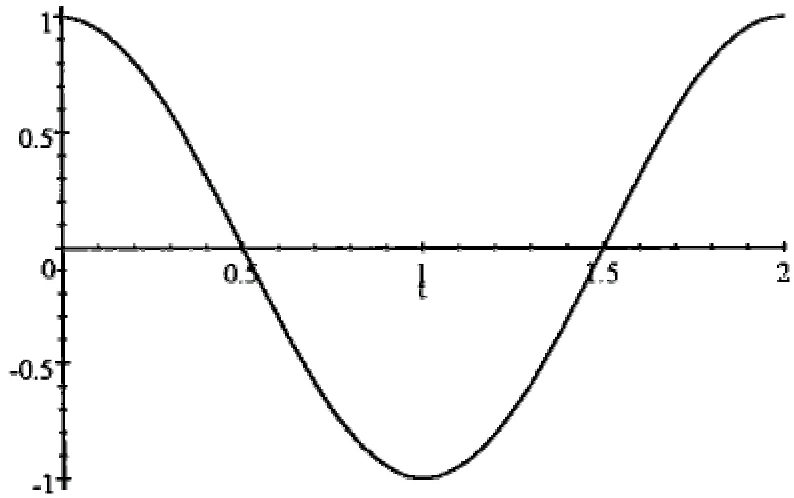
- I want the depth of understanding that can only come only from figuring things out from first principles,
- "peeking at the answer" from someone else's work would spoil the fun for me,
- I hope to give a somewhat fresh outlook on things.

In 1990, one of my fellow autocrossers asked me to write a monthly column for the SCCA CalClub newsletter. After receiving lots of encouragement, I released the columns to the Internet via Team Dot Net. Back then, the Internet was really small, so I was just sharing the articles in a convenient way with other autocrossers. Since then, the Internet got big and my articles have acquired a life of their own. I have received thousands of happy-customer emails from all over the world, plus a few hate mails (mostly about article #4, in case you're wondering).

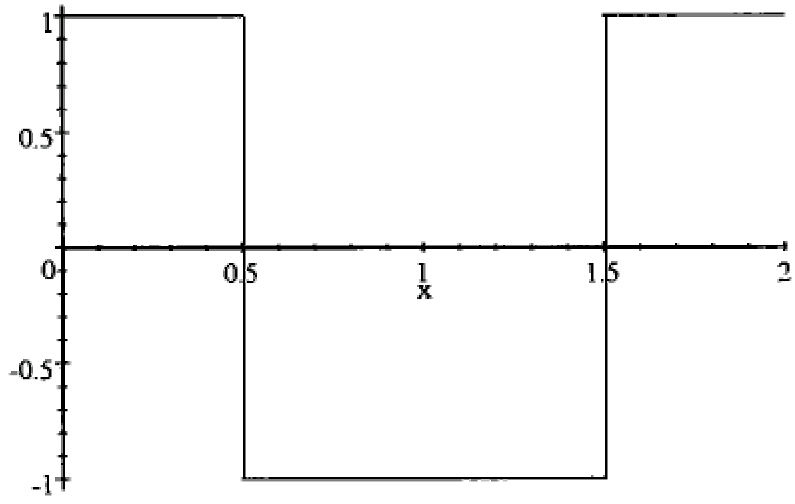
So, here we go again. This month, I'd like to understand, from first principles, why it's so important to be smooth on the controls of a racing car. To me, "smooth" means avoiding jerkiness when applying *or releasing* the brakes, the gas, or steering. Most of the time, you want to roll on and off the gas, squeeze on and off the brakes, slither in and out of steering. It's just as important to avoid jerkiness at the end of a manoeuvre as at the beginning. For example, when steering, not only should you start turning the steering wheel with a gradual, smooth push, but you want to complete the wind-up with a gradual, smooth slowing of the push. Likewise, when you unwind the wheel, you want to start and stop the unwinding smoothly. Thus, a complete steering manoeuvre consists of *four* gradual, slithery start-and-stop mini-manoeuvres. A complete braking event has four little mini-slithers: one each for the start and stop of the application and the releasing of the pedal. Same for the throttle.

Ok, great, but why? At first blush, it seems one would be able to get back on the gas *more quickly* by snapping the throttle on or get into a corner more quickly by whipping the wheel. Furthermore, supposing we can justify smoothness, are there exceptions to the rule? Are there times when it is best to snap, whip, or jerk? And exactly how smooth should one be? Smoothing means slowing the control inputs down, in a particular way, so it's obviously possible to be too smooth, as in not quick enough, as in not getting as much out of the car as it's capable of delivering.

Let's tackle "why", first. As usual, physics has technical meanings for everyday words. One of the "physically correct" meanings of "smooth" is *sinusoidal*. A sinusoid is a curve that looks like this:



If we think of, say, steering-wheel winding angle as proportional to the vertical axis and time in seconds along the horizontal axis, then this picture describes a really smooth windup taking one second followed immediately by a really smooth unwinding taking another second. In fact, you can easily see the four mini-slithers discussed above as the head-and-tail-sections of the bumps and valleys of the curves. So, the question "why", in technical terms, amounts to asking why such a curve represents better steering input than a curve like the following, "upside-down-hat" curve:



Now, here's the reason: sinusoidal inputs are better because they match the natural response of the car! The suspension and tyres perform, approximately, as *damped harmonic oscillators*, or DHOs. A DHO can be in one of three conditions: *underdamped*, *critically damped*, or *overdamped*. In the underdamped condition, a DHO doesn't have a strong *damper*, which is another term for shock absorber. An underdamped DHO responds sinusoidally. We've all seen cars with broken shocks bouncing up and down on the springs. In the critically damped and overdamped conditions, the car bounces just once, because the damper provides some friction to quiet down continued bouncing. However, even in these conditions, the one bounce has an approximate sinusoidal shape.

The most important parameter of any DHO is its *frequency*. In the underdamped condition, the frequency corresponds to the number of bounces per second the DHO performs. In the critically damped and overdamped conditions--as well as in the underdamped condition, the frequency corresponds to the *resonance* frequency or natural frequency of the system! In other words, if you provide so many inputs per second, back and forth, as in a slalom, at the resonance frequency, the car will have maximal response. If the inputs are faster, they will be too fast for the DHO to catch up and rebound before you've reversed the inputs. If the inputs are slower, the DHO will have caught up and started either to bounce the other way or to settle, depending on condition, when the reverse input comes in.

So here's the bottom line: to maximize the response of a car, you want to provide steering, braking, and throttle inputs with sinusoidal shapes at the resonance frequency of the DHOs that constitute the suspension and tyre systems. Inputs that are more jerky just dump high-frequency energy into the system that it must dissipate at lower frequencies. In other words, jerky inputs *upset* the car, which what drivers say all the time. By matching the shape and frequency of your control inputs to the car's natural response curve, you're telling the car to do something it can actually do. By giving the car an "instruction" like the upside-down hat, you're telling it to do something it can't physically do, so it responds by flopping and bouncing around some approximation of your input. Flopping and bouncing means not getting optimum traction; means wasting energy in suspension oscillation; means going slower. Now, there is an exception: if the front tyres are *already* sliding, a driver may benefit from quickly steering them into line, hoping to "catch" the car. Likewise, a jerky blip on the throttle with the clutch engaged to bring up the revs to match the gears on a downshift is usually the right thing to do. But, when the car is hooked up, getting the most out of the car means *simulating* the response of the various DHOs in the system with steering, braking, and throttle inputs.

Now we know the physics behind it. Let's do some math!

The frequency turns out to be $\omega = \pm\sqrt{k/m}$, as we show below. k is the *spring constant*, typically measured in pounds per inch, and m is the mass of the sprung weight, typically measured in pound-masses. Suppose our springs were 1,000 lb/in, supporting about 800 lb of weight on one corner of the car. First, we note that a pound *force* is roughly (1/32) *slug* - ft/s^2 and that a pound *weight* is (1/32) *slug*. So, we're looking at

$$\omega = \pm\sqrt{\frac{1{,}000\dfrac{\text{lb}-\textit{force}}{\text{in}}12\dfrac{\text{in}}{\text{ft}}\dfrac{1}{32}\dfrac{\textit{slug}-\text{ft/s}^2}{\text{lb}-\textit{force}}}{\dfrac{800\,\text{lb}-\textit{weight}}{32\,\text{lb}-\textit{weight/slug}}}}$$

$$= \pm\sqrt{\frac{12{,}000}{800\,\text{s}^2}} = \pm\sqrt{\frac{120/8}{\text{s}^2}} = \pm\sqrt{\frac{15}{\text{s}^2}} \approx 4/\text{s}$$

Notice that we've used the back-of-the-envelope style of computation discussed in part 3 of this series. We've found that the resonance frequency of one corner of a car is about 4 bounces per second! This matches our intuitions and experiences: if one pushes down on the corner of a car with broken shocks, it will bounce up and down a few times a second, not very quickly, not very slowly. We can also see that the frequency varies as the square root of the spring constant. That means that to double the frequency, say, to 8 bounces per second, we must quadruple the spring strength to 4,000 lb/in or quarter the sprung weight to 200 lb. [Note added in proof: My friend, Brad Haase, has pointed out that 4 Hz, while in the "ballpark", is much too fast for a real car. Now, this series of articles is only about fundamental theory and ballpark estimates. Nonetheless, he wrote convincingly "can you imagine a 4-Hz slalom?" I have to admit that 4 Hz seemed too fast to me when I first wrote this article, but I was unable to account for the discrepancy. Brad pointed out that the suspension linkages supply leverage that reduces the effective spring rate and cited the topic "installation ratio" in Milliken's book *Race Car Vehicle Dynamics*. Since I have not peeked at that book, on purpose, as stated in the opening of this entire series and reiterated in this article, I can only confidently refer you there. Nonetheless, intuition says that 1 Hz is more like it, which would argue for an effective spring rate of 1000 / 16 = 62 lb/in.]

How do we derive the frequency formula? Let's work up a sequence of approximations in stages. By improving the approximations gradually, we can check the more advanced approximations for mistakes: they shouldn't be too far off the simple approximations. In the first approximation, ignore the damper, giving us a mass block of sprung weight resting on a spring. This model should act like a corner of a car with a broken shock.

Let the mass of the block be m. The force of gravitation acts downwards on the block with a magnitude mg, where $g = 32.1$ ft/s^2 is the acceleration of Earth's gravity. The force of the spring acts upward on the mass with a magnitude $k(y_0 - y)$, where k is the spring constant and $(y - y_0)$ is the height of the spring above its resting height y_0 (the force term is positive – that is, upward – when $y - y_0$ is negative – that is, when the mass has compressed the spring and the spring pushes back upwards). We can avoid schlepping y_0 around our math by simply defining our coordinate system so that $y_0 = 0$. This sort of trick is very useful in all kinds of physics, even the most advanced.

It's worth noting that the model so far ignores not only the damper, but the weight of the wheel and tyre and the spring itself. The weight of the wheel and tyre is called the *unsprung weight*. The weight of the spring itself is partially sprung. We don't add these effects in the current article. Today, we stop with just adding the damper back in, below.

Newton's first law guides us from this point on. The total force on the mass is $-ky - mg$. The mass times the acceleration is $m(dv_y/dt) = m(d^2y/dt^2)$, where v_y is the up-and-down velocity of the mass and dv_y/dt is the rate of change of that velocity. That velocity is, in turn, the rate of change of the y coordinate of the mass block, that is, $v_y = (dy/dt)$. So, the acceleration is the *second* rate of change of y, and we write it as d^2y/dt^2 because that's the way Newton and Leibniz first wrote it 350 years ago. We have the following *dynamic equation* for the motion of our mass block.

$$F = ma \Rightarrow m\frac{d^2y}{dt^2} = -ky - mg$$

Let's divide the entire equation by m and rearrange it so all the terms are on the left:

$$\frac{d^2y}{dt^2} + \frac{k}{m}y + g = 0$$

If we're careful about units, in particular about *slugs* and lbs (see article 1), then we can note that k/m has the dimensions of 1/sec^2, which is a frequency squared. Let's define

$$\omega^2 = \frac{k}{m}$$

yielding

$$\frac{d^2y}{dt^2} + \omega^2 y + g = 0$$

We need to solve this equation for $y(t)$ as a function of time t. To follow the rest of this, you'll need to know a little freshman calculus. Take, as *ansatz*,

$$y = A + Be^{C\omega t}$$

then

$$\frac{dy}{dt} = C\omega Be^{C\omega t} = C\omega(y - A)$$

and

$$\frac{d^2y}{dt^2} = C\omega\frac{dy}{dt} = (C\omega)^2(y - A)$$

therefore

$$\begin{aligned}\frac{d^2y}{dt^2} + \omega^2 y + g &= (C\omega)^2 y - A(C\omega)^2 + \omega^2 y + g \\ &= \omega^2(C^2 + 1)y - (A(C\omega)^2 - g) \\ &= 0 \text{ iff } C^2 = -1 \text{ and } A = -g/\omega^2\end{aligned}$$

So, we see there are two solutions, $y(t) = A + B_1 e^{i\omega t}$ and $y(t) = A + B_2 e^{-i\omega t}$. In fact, the time-dependent parts of these solutions can operate simultaneously, so we *must* write $y(t) = A + B_1 e^{i\omega t} + B_2 e^{-i\omega t}$ in all generality. The values of the two unknowns B_1 and B_2 are determined by two initial conditions, that is, the value of $y_0 = A + B_1 + B_2$ and $(dy/dt)(0) = i\omega(B_1 - B_2)$.

Let's get out of the complex domain by writing

$$\begin{aligned}B_1 e^{i\omega t} + B_2 e^{-i\omega t} &= B_1(\cos\omega t + i\sin\omega t) + B_2(\cos\omega t - i\sin\omega t) \\ &= (B_1 + B_2)\cos\omega t + i(B_1 - B_2)\sin\omega t \\ &\overset{\Delta}{=} C_1\cos\omega t + C_2\sin\omega t\end{aligned}$$

This definition makes our initial conditions simpler, too:

$$y(0) = C_1;\ \upsilon_y(0) = \omega C_2$$

It's easy, now, to add the damper. Damping forces are proportional to the velocity; that is, there is no damping force when things aren't moving. Each corner approximately obeys the equation

$$\frac{d^2y}{dt^2} = -\frac{\delta}{m}\frac{dy}{dt} - \frac{k}{m}y - g$$

where δ is the damper response in $\text{lb - force} / (\text{ft} / \text{s})$. The three rightmost terms represent forces, and they are all negative when y and dy/dt are positive. That is, if you pull the sprung weight up, the spring tends to pull it down. Likewise, if the sprung weight is moving up, the damper tends to pull it down. The force of gravitation always pulls the weight down. Let's rewrite, as before:

$$\frac{d^2y}{dt^2}+\gamma\frac{dy}{dt}+\omega^2y+g=0$$

where $\omega^2 = k/m$ and $\gamma = \delta/m$. If, as before,

$$y = A + Be^{C\omega t}$$

then

$$\frac{dy}{dt} = C\omega Be^{C\omega t} = C\omega(y - A)$$

and

$$\frac{d^2y}{dt^2} = C\omega\frac{dy}{dt} = (C\omega)^2(y-A)$$

therefore

$$\begin{aligned}\frac{d^2y}{dt^2}+\gamma\frac{dy}{dt}+\omega^2y+g &= (C\omega)^2(y-A)+C\gamma\omega(y-A)+\omega^2y+g\\ &= ((C^2+1)\omega+C\gamma)\omega y-(AC\omega(C\omega+\gamma)-g)\\ &= 0 \text{ iff } \omega C^2+\gamma C+\omega=0 \text{ and } AC\omega(C\omega+y)=g\end{aligned}$$

You may remember the little high-school formula $(-b\pm\sqrt{b^2-4ac})/2a$ for the solution of a quadratic equation. This gives us the answer for C:

$$C=\frac{-\gamma\pm\sqrt{\gamma^2-4\omega^2}}{2\omega}$$

and I'll leave the simple arithmetic for A and the initial conditions to the reader. The critically damped condition obtains when $\gamma = 2\omega$, overdamped when $\gamma > 2\omega$, and underdamped when $\gamma < 2\omega$. In the underdamped condition, C has an imaginary component and the exponentials oscillate. Otherwise, they just take one bounce and then settle down.

It will be fun and easy for anyone who has followed along this far to plot out some curves and check out my math. If you find a mistake, please do let me know (I just wrote this off the top of my head, as I always do with these articles).

We could improve the approximation by writing down the coupled equations, that is, treating all four corners of the car together, but that would just be a lot more math without changing the basic physics that the car responds more predictably to smooth inputs and less predictably to jerky inputs. Another improvement would be to add in the effect of the unsprung and partially sprung weight.

The Physics of Racing, Part 15: Bumps In The Road

Brian Beckman PhD,
and Jerry Kuch

This month, we investigate how the effects of road bumps vary with speed. Everyone has experienced that bumps are more punchy as speed increases. A bump that you barely notice at 50 mph can sting at 100 mph. But what about at 200 mph? Will it just smack a little harder, or will it knock your teeth out or, worse, cause you to lose control? Could a bump be the limiting factor in cornering speed? In an aerodynamic car, could a bump cause a sudden and catastrophic loss of downforce and adhesion? To analyse such things, we need an understanding of the variation of bump violence with speed.

At the expense of a little storytelling, let's explain how this topic came up. In particular, where is an amateur motorhead going to have to worry about bumps at 200 mph? At autocrosses, speeds are low, by design, to give everyone a safe venue to challenge the limits. If you're going to spin out, an autocross is the place to do it. Low speed also means, though, that bumps, unless very severe, aren't dominant. On a road course, speeds are higher, as are the consequences of losing control. But speeds are not higher *everywhere*, not for extended times, and seldom approach 200 mph. There are two commonplace scenarios with extended time at high speeds: oval courses and open-road racing. High-speed oval racing is a specialized sport not often encountered by amateurs. Since the focus of this series is on grassroots, amateur hijinks, we'll look at open-road racing.

In Part 11 of this series, we took a scenario for braking from 200 mph from the Silver-State Challenge (SSC) in Nevada. My co-author, Jerry Kuch, and I just ran the 2000 Nevada Open Road Challenge (NORC). This is the May version of the SSC, which is held in September. In all other regards, the NORC and the SCC are the same. For most of the 230 cars entered, these are high-speed, time-speed-distance (TSD) rallies. In each of the sixteen TSD classes, the car running as close as possible to the target speed, over or under, wins. There are TSD classes every five mph from 95 to 170 inclusive, with high and low breakout speeds set by safety concerns. There is also an Unlimited, non-TSD class, in which fastest car wins. This May, the winner of Unlimited averaged 207 mph over a ninety-mile distance and another Unlimited car posted a top speed of 227 mph. Jerry and I ran in the 130-mph class with a top speed of 165 mph.

The SCC and NORC run on a ninety-mile stretch of highway 318 from Lund to Hiko in the Nevada outback, roughly along the shortest path from Twin Falls, ID to Las Vegas. The course runs from north to south, and the road is fabulously stark and beautiful in the unique way of remote desert roads. One is humbled by the realization that if stranded, one would surely perish, probably in a few hours' time, from heat exhaustion, exposure, and dehydration. It's great.

Hwy 318 events have been run continuously on since 1988. In 1990 and 1991, Mark Thornton, a fellow autocrosser, built up his 1986 Super Stock corvette into a Nevada car. Mark and I had nearly identical SS 'vettes, and we often swapped cars at autocrosses. These cars happened to be almost the same as the famous yellow 'vette that Roger Johnson, of multiple SCCA National Championships, still runs in SS, if I'm not mistaken. I know that Roger has driven my car, and I can't recall whether he ever drove Mark's, but I did, many times.

Mark, now deceased, was a bit of a bad boy, and Hwy 318 had just the kind of cachet that appealed to him. The legend goes that the events had been organized by the survivors of the old, illegal 'cannonball' runs. Of course, the NORC and SCC are properly sanctioned and completely legal, despite the fact that they use temporarily closed public highways rather than dedicated race courses.

Not content to play in the TSD classes, Mark decided to convert the black car into an Unlimited machine. I was with Mark when he handed his car off to Dick Guldstrand for blank-check suspension work, and I was in the loop when it went to John Lingenfelter for a reliable engine capable of 200 mph. I met up with Mark in Las Vegas to help with the final preparation of the car. I took a few, tyre-warming hops in the car, and, with nearly 600 HP, I can tell you it was seriously fast. Feel free to check out the car's specs at http://www.angelfire.com/wa/brianbec/foober.htm.

Unfortunately, on race day, the car had an oil fire in the first, six-mile straightaway, due to the headers' being a bit too close to the oil-filter canister. The required, on-board halon system saved the car and Mark and I saved what residual fun we could putting it back together and trailering it home. Later that year, Mark won a Triathlon of Motorsports hosted by a hotrodding magazine in the car, and, if I'm not mistaken, repeated the feat in '92. I have been told the car was featured on the cover of the magazine somewhere in those two years, but I have not checked that myself.

I moved to Washington State and lost touch with Mark, who had a non-motorsports accident and passed away. Mark was not uniformly liked, but even his detractors will grant that he was a truly gifted driver and an engaging, entertaining, complex character. Many, currently active autocrossers will remember him.

By sheer, stupid luck, I stumbled across Mark's Nevada car for sale in Florida in 1999. This is about as far away from Seattle as one can get, but the kismet was too much to ignore. I had driven this car many times in anger, had crewed it, was friends with its creator. It just had to come home to me, didn't it? Furthermore, it just HAD to run again in Nevada, didn't it?

I bought the car and began the complex job of preparing it for NORC. One does not contemplate running 200 mph without giving a car a complete check-up. The energy available for destruction at 200 mph is four times the energy available at 100 mph, and sixteen times that available at 50 mph. Furthermore, the car had had an active, open-track life in the intervening years and it was time to tear it down and check it all out. You do NOT want an engine to seize or a suspension part to break at 100 mph, let alone at 200 mph.

With two months to spare, it became obvious that the car would not be ready in time. Better safe than sorry, I asked the mechanics not to hurry and to make sure the car is done *right*. The standards for mechanical work on high-speed cars must be significantly higher than it is for road-going and autocross vehicles, for safety. The standards should be comparable to those in aviation. Hurrying is a recognized no-no in aviation, and I applied the same logic to the car work. As I write, I have an ultimate goal of running it in SCC and NORC in ‘01 and ‘02.

I had already committed to run the ‘00 NORC, so I slapped a roll cage in my ‘98 Mallett 435 and went on down. This is another fabulous vehicle, but I hadn’t intended to run it in high-speed events until the last minute. It was quite a hustle to get the required safety gear properly installed in time. In hindsight, I don’t regret the decision. The car really came to life at NORC and I’ve run it in several high-speed events since then.

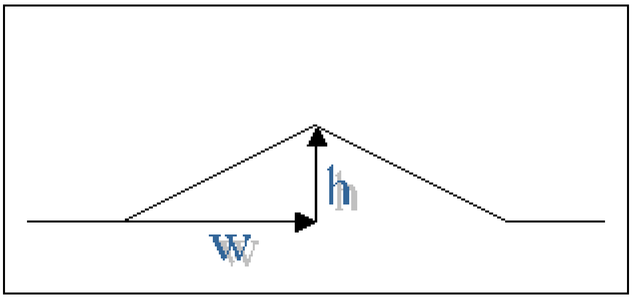
Our flight plan called for holding speeds up to 165 for minutes at a time. As part of planning, we did a survey and calibration run of the course at legal, highway speeds. On the survey run, we noticed several bumpy spots. Driving over them at 70 mph, they were not frightening. But, we had to figure out what to expect at 165. So, right there in the middle of nowhere, we whipped out some envelopes, turned them over, pulled multicolour pens from our pocket protectors, and started scribbling. Geek racing at its best.

Let us take a moment to review the goals and methods of the “back-of-the-envelope” (BOE) style of analysis introduced in Part 3 of this series. Frequently, one simply needs a ballpark estimate or a trend. These are often *much* easier to get than are detailed, precise answers. In fact, they are often easy enough that they can be literally scribbled out on the backs of envelopes *in the field*. And that’s the key point: we needed a rough idea of how the violence of the bumps varies with speed, and we needed it right then and there in the field.

Another benefit of the BOE style is that it can give one a quick plausibility check on numerical data back at the lab. Thoroughgoing engineering analysis usually entails dozens of interlocking equations solved on a computer resulting in tables, plots, and charts. The intuition gets lost in the complexity. It’s sometimes impossible to say, just by looking at a table or chart, whether the results are correct. On the other hand, to get our BOEs, we often make very gross approximations, such as treating the car as a rigid body; or ignoring its track width, that is, treating it as infinitely thin; or ignoring the suspension altogether; or even treating the whole car as a point mass, that is, as if all its mass were concentrated at a single point. Even so, the results are often not wildly off the numerical data, and the discrepancies can usually be explained via non-quantitative arguments. If the BOE and numerical results *are* wildly different, then some detective work is indicated: one or both of them is probably wrong.

BOE is really a semi-quantitative oracle to the physics. These articles are about the physics of racing as opposed to the engineering of racing. We’re primarily interested in the fundamental, theoretical reasons for the behaviour of racing cars. The trends and ballpark estimates we get from BOEs often do the job. Of course, this doesn’t mean we won’t get into more detailed treatments and computer simulation. It’s just that we will always be focusing on the physics.

All that said, as usual for BOE, we start with a simplistic model we can solve easily. Think of a bump in the road as a pair of matched triangles, one leading and one trailing.



Let the width of each triangle be $\boldsymbol{w}$ and the height be $\boldsymbol{h}$. Suppose a car approaches the bump with horizontal speed $\boldsymbol{v}$. To assess the violence of the bump, let's ask what vertical acceleration the car will experience? If we assume a simplistic model of the car as a rigid body, we get an instantaneous, infinite acceleration right at the instant the car contacts the rising edge. We get further infinite, vertical accelerations at the two other cusps of bump the geometry. However, we know that the tyres and suspension will smooth out these sudden impulses. Calculating the effects of tyre and suspension flex is too time-consuming to do in the field even if we had data and computers on hand. However, we can get a useful approximation by assuming that the acceleration is distributed over the entire bump.

If the bump is shallow $(\boldsymbol{h} \ll \boldsymbol{w})$ and the car is fast, then the horizontal speed doesn't change very much and the car goes up the leading edge of the bump in time $\boldsymbol{t} = \boldsymbol{w} / \boldsymbol{v}$. In that time, the car goes upward a distance $\boldsymbol{h}$, thereby acquiring a vertical speed of $\boldsymbol{v}_y = \boldsymbol{h} / \boldsymbol{t} = \boldsymbol{v}\boldsymbol{h} / \boldsymbol{w}$. Since it acquires that velocity, very roughly, in time $\boldsymbol{t}$, we can estimate the vertical acceleration to be

$$\boldsymbol{a}_y \approx \boldsymbol{v}_y / \boldsymbol{t} = \boldsymbol{h} / \boldsymbol{t}^2 = \boldsymbol{v}^2 \boldsymbol{h} / \boldsymbol{w}$$

Uh oh. BOE says that the severity of a bump goes up as the *square* of the speed. A bump you can feel at 50 mph is going to be *sixteen times* worse at 200 mph and will most definitely get your attention. The little whoopdeedoos we were noticing at 70 mph would feel $(165/70)^2$ = 5.5 times worse at our planned speed: definitely something to anticipate on-course before we hit them. This BOE also says that the nastiness varies inversely as the width. The wider the bump, the less nasty, linearly. This is plausible.

Now, let's refine the analysis a little. Conservation of energy dictates that the horizontal speed of the car must change. In our simplified, two-dimensional BOE, the velocity vector, $\vec{\boldsymbol{v}}$, consists of two components, horizontal speed, $\boldsymbol{v}_x$, and vertical speed, $\boldsymbol{v}_y$. These quantities obey the equation

$$|\vec{\boldsymbol{v}}|^2 \stackrel{\Delta}{=} \boldsymbol{v}^2 = \boldsymbol{v}_x^2 + \boldsymbol{v}_y^2$$

whether on the flat or on the bump, that is, no matter what the inclination of the road. We've presupposed, here, that *vertical* always means "in the direction of Earth's gravitation." If we do not change the kinetic energy of the moving car, then $\frac{1}{2}\boldsymbol{mv}^2$ stays constant, therefore $\boldsymbol{v}^2$ stays constant. On the leading-edge ramp of the bump, remembering trigonometry,

$$\boldsymbol{v}_x = \boldsymbol{v}\cos(\arctan(\boldsymbol{h}/\boldsymbol{w})) = \boldsymbol{vw}/\sqrt{\boldsymbol{h}^2 + \boldsymbol{w}^2}$$

$$\boldsymbol{v}_y = \boldsymbol{v}\sin(\arctan(\boldsymbol{h}/\boldsymbol{w})) = \boldsymbol{vh}/\sqrt{\boldsymbol{h}^2 + \boldsymbol{w}^2}$$

Define, as shorthand, $\boldsymbol{r} \equiv \sqrt{\boldsymbol{h}^2 + \boldsymbol{w}^2}$, yielding $\boldsymbol{v}_x = \boldsymbol{vw}/\boldsymbol{r}$, $\boldsymbol{v}_y = \boldsymbol{vh}/\boldsymbol{r}$. Using the same approximation as above, we assume that we acquire a vertical velocity of $\boldsymbol{v}_y$ in time $\boldsymbol{t} = \boldsymbol{w}/\boldsymbol{v}_x = \boldsymbol{wr}/\boldsymbol{vw} = \boldsymbol{r}/\boldsymbol{v}$, for a vertical acceleration of

$$\boldsymbol{a}_y \approx \frac{\boldsymbol{v}_y}{\boldsymbol{t}} = \frac{\boldsymbol{vh}/\boldsymbol{r}}{\boldsymbol{r}/\boldsymbol{v}} = \frac{\boldsymbol{v}^2\boldsymbol{h}}{\boldsymbol{r}^2} = \frac{\boldsymbol{v}^2\boldsymbol{h}}{\boldsymbol{h}^2 + \boldsymbol{w}^2}$$

This still varies as the square of the speed, we just take a little more time to go over the bump. The only difference to the prior formula, $\boldsymbol{v}^2\boldsymbol{h}/\boldsymbol{w}$, is the appearance of $\boldsymbol{h}^2$ in the denominator.

Consider the case of a high, narrow bump. This case was not covered by our first BOE, which assumed that $\boldsymbol{h} << \boldsymbol{w}$. Now, with a high bump, $\boldsymbol{h}^2 >> \boldsymbol{w}^2$ and $\boldsymbol{a}_y \approx \boldsymbol{v}^2/\boldsymbol{h}$, meaning that the severity of the bump will go *down* linearly with increasing height. Within the confines of our model, this makes sense, because a higher bump gives the car a greater vertical distance in which to suffer its increased vertical velocity, but this doesn't seem *intuitively* correct. A higher bump should be nastier, shouldn't it?

Furthermore, of course, at constant throttle, the kinetic energy of the car *will* change because the force of gravitation will attenuate the vertical velocity. So, in our next consultation of the BOE oracle, we must reduce $\boldsymbol{a_y}$ by

$$\boldsymbol{g} \approx 32\frac{\text{ft}}{\text{s}^2}$$

The bump is getting less nasty all the time, and it's obvious that we're hitting the limitations of this BOE analysis. To expose the limitations even more starkly, consider two more questions: (1) what about the trailing edge? and (2) what about depressions, that is, down-bumps?

As to the trailing edge, a simplistic car-as-rigid-body would just launch ballistically from the top of the bump. Of course, in a real car, tyre elasticity and the suspension would endeavour to keep the tyres on the ground. Short of launching, there would just be weight loss causing rebound of the tyre sidewalls and the suspension springs.

Nevertheless, everyone knows that a ballistic projectile assumes a parabolic flight path, so, as long as the parabola off the top of the bump remains vertically above the down-ramp, our car-as-rigid-body is assured of taking to the air. With the simple bump geometry, we can see that a parabolic launch *always* starts off above the trailing-edge triangle. It intersects the road again either somewhere on the down-ramp or on the following flat bit of road, depending on horizontal speed.

As to a depression – a down-bump as opposed to an up-bump – a car-as-rigid-body will simply have a ballistic phase before suffering an upward acceleration. At this point, I think we've reached the point of diminishing returns. Let us first repeat that the BOE style is doing what it's supposed to do: getting us rough trends and quantities in the field. Primarily, we wanted to find out how bump severity varies with speed, and we've got our answer: roughly quadratically. We are seeing some ways in which the model departs from intuition and reality and it's time to think about how to improve it back at the lab.

The first point to notice is that we drew a pair of triangles for our bump, but used them only to calculate the time to traverse the bump and the height acquired over that time. This is not a proper *dynamic* analysis, in which we would use Newton's laws to model the motion of the car up and down the bump. At a glance, one can distinguish a dynamic analysis by the presence *mass* in the equations. Nowhere did we use the mass of the car in our BOEs above. Dynamic analysis is often too hard to do in the field because it involves integrating differential equations, almost always by computer.

Another problem concerns our simplistic bump geometry. As noted above, strictly speaking, the severity of a bump on a rigid body *infinite*, no matter what the speed. The reason is that the car acquires its vertical component of velocity instantaneously - in zero time - upon hitting the bump, so the rate of change of the vertical velocity, that is, the vertical acceleration, is infinite at the instant the bump is encountered, then zero on the body of the up-ramp.

Our list-of-things-to-do, should we wish to improve the model, includes the following tasks:

Model the geometry of the bump more carefully, accounting for the fact that the initiation of the up-ramp, no matter how severe, cannot, in fact, be mathematically instantaneous. Draw some sort of little sinusoidal or exponential curves to account for the actual road profile.

Integrate the equations motion of the car over the bump.

Model the car more carefully, accounting for tyre flexion, springs, shocks, suspension geometry, mass distribution, moment of inertia, and all the rest. This will entail designing a suspension.

These improvements put us squarely back in the lab. Ultimately, we will resort to computer simulation. As promised years ago, that is the ultimate goal of this series of articles: to spec out a simulation program. Better late than never, right?

Note on Part 14, *Why Smoothness* The last episode of the *Physics of Racing* sparked a debate on reasonable values for effective wheel spring rates and raised the notion of "installation ratio." The particular point raising the debate was whether 4 Hz was a reasonable value for the resonance frequency of a real racing chassis. It seems it is certainly too fast for a road-going car, however, in the time since Part 14 was released I was introduced to a 1980 Group C Ferrari Sports Car. This is essentially a Le Mans car with a lower horsepower engine, for reliability. It is a fully aerodynamic car with ground effects that corners at 2.7g and brakes at 4g. Here's the kicker: its ride height is about half an inch, it does NOT bottom out on bumps, and its spring rate is 14,000 lb/in [sic]. I don't know the installation ratio for this car, but I would be surprised if its chassis resonance frequency was not on the order of 4 Hz or even higher.

The Physics of Racing, Part 16: RARS, A Simple Racing Simulator

Brian Beckman, PhD

If you've been following this series, you know that I've been moving inexorably toward a computer simulation of racing. I've repeatedly debated with myself writing a new one completely from scratch versus starting with someone else's work. Ten years ago, when I started this series, the choice was easy. Since there was nothing out there, I had to start from scratch. The situation has changed. There is at least one competently executed program in the public domain.

Doing a derivative work has undeniable advantages, but conflicts with one of the enduring goals of this series, that is, to do totally original work rather than to recapitulate information you can get from other sources. However, one has to start somewhere. After all, it would be silly for me to rediscover Newton's laws, so I take those as given. Likewise, I've concluded that it would be silly for me to invent the *infrastructure* for a simulation. It would be a very long digression indeed from the Physics of Racing to cover all the groundwork such as

- memory management, windowing, graphics, rendering, data reporting, etc.
- programming languages, scripting, object technology
- simulation technology: time-stepping, eventing, dynamics solvers
- data structures for track description and cars
- arbitrary choices for coordinate systems

All this, while interesting, is not physics. Furthermore, nowadays, it's all more-or-less conventional technology. It's not terribly important for us to make choices in these domains if we can find a competent base platform in which reasonable choices have already been made.

So, with only a little reluctance, I take the decision to start with an existing program. My choice is RARS, the Robot Auto Racing Simulator. This is a lovely, surprisingly simple platform for programmers to experiment with robotics. Its purpose is to support distributed virtual racing competitions, in which entrants write robot drivers and enter them in planned events. The last competitions I have been able to find on the web were conducted in 1999. It is *not* a high-fidelity simulation, and, in fact, was never intended to be. Its physics is simplified in a very clever way to make the main challenge for competitors the writing of robots rather than struggling with elaborate, high-fidelity physics. It supplies a working infrastructure and a large amount of decent data describing famous tracks. Finally, so far as I can tell, RARS is in the public domain.

The simplifications in RARS make it the perfect starting point for enhancing the *physics* without having to reinvent the peripheral aspects of a simulation program. Note that RARS was *designed* for public contribution: the program was originally made to be easy to modify. The usual mode of modification is for competitors to add new robots. However, it is just as easy to change the physics, as I intend to do. Now, as I take the program in new directions, I will either have to modify the robots or, possibly, create a new, public racing series and throw open the writing of new robots to everyone. Only time will tell what works out best. As usual, however, I will make changes *incrementally*, never deviating very much from the working base. This strategy will not only keep the changes under control, but also enable me to explain to you what's going on, step-by-step.

Therefore, I will create a copy of the sources and change the name of my copy to RARSEP, for "RARS, Enhanced Physics". I will post the source code of my changes on the web to keep the new project rolling along.

My first, long-term goal with RARSEP is to ***find optimal racing lines***. In particular, I need a way to answer questions about racing lines, such as whether the shortest line or the highest-speed line around a particular feature results in the lowest time around the entire course, that is, with the feature in context. Such a question is part of "reading a course", one of the tasks of every racer. In practice, this is a trial-and-error process involving folklore, experience, and experimentation.

For instance, at a recent track day I attended, two instructors, each with many hours on this particular course, were debating a certain combination of slow corners. After quite a bit of haggling and white-board hacking, they agreed that the classic line they *had* been taking for years was probably not the fastest line. It will warm the hearts of autocrossers to find that they had discovered that the autocrosser line, rather than the class road-racer line, was probably fastest.

Autocrossers spend most of their effort finding the fastest way around *slow* corners, whereas the primary challenge for road-course drivers is finding the fastest way around *fast* corners. There is no end of reading material supporting the *classic,* road-racing lines: enter as wide as possible (or, as *high*, as one would say in NASCAR), trail-brake, get back on the throttle in the first half, squeeze on the gas, look up, late apex, and track out. As often as not, however, autocrossers find that simply hugging the inside as tightly-as *low*-as possible yields the quickest way around. Why? Is there science behind this? Can they both be right? How about both wrong? What about intermediate cases: medium-slow and medium-fast corners?

That is an example of the *kind* of question that we want to answer with a simulation program. It was interesting that the instructors' debate concerned slow corners. No one was debating that the fast corners should be taken classically. But, and here I hypothesize, slow corners have the characteristic that ***the corner is not very much larger than the car***. Could it be that when this is true the classic racing line is suboptimal? Experienced autocrossers would, when coming up on such corners, without even thinking about it, go in low and tight and just carry speed or toss-and-catch the car. The instructors had, following the classic theory, been going in high and wide, turning in late, and thereby wasting time trying to form a classic line around corners not much bigger than the car. But, could it be that the classic theory is not best when a corner is so small that the wheelbase of the car is a significant fraction of the distance around the corner? Maybe there are other factors, though. Could it be that the size of the corner does *not* suffice to distinguish an “autocrossy” corner from a “road-racey” corner? Does context matter, as in whether the corner is near other corners or near straights?

It seems that even the most experienced drivers of a particular track will occasionally discover improvements to the line. Some of these improvements depend on transient conditions like weather or the particulars of a certain car or setup. Lots of tracks have canonical “rain lines” that differ from the “dry lines”. I would also bet that Winston Cup cars take different lines around Sears Point and Watkins Glen than do ground-effects sports cars and downforce formula cars. But some improvements will be deep, permanent, invariant revelations that may have eluded the racer on previous outings and analysis. That, in a nutshell, is the first place we’re going with RARSEP: to have a way to answer such questions.

I will start with the Windows port of RARS version 074, which you can get in source and binary forms from the following web sites:

- http://users.skynet.be/mgueury/rars/rars.html
- http://www.cgr.ki.se/cgr/persons/mremm/rars/main.htm

I choose the Windows port because it’s most convenient for me: I already have working development systems on Windows, whereas to work on other platforms would entail ramp-up time and money. The RARS code base is currently portable to multiple platforms, including Linux and Windows. The code is very well partitioned, so that the platform-dependent bits are separated from the platform-independent bits. Everything I intend to do will be in the platform-independent parts of the program and should build without difficulty on all the platforms. However, I will not be able to test my changes on all platforms - the Physics of Racing is not an exercise in industrial-strength, portable software development. While I have no intention of making non-portable changes, there is a small risk that I might inadvertently do so and it could happen some files might someday need a little tweaking to get going on other platforms. I am sure my readers will let me know about it.

The web sites contain very complete descriptions of how to build and run the program, plus how to write robots. To write a robot, one needs to understand the existing physics model of RARS. Similarly, to enhance the physics, we’ll need to know the same thing. It presently appears that the best way to enhance the physics incrementally will be in the context of writing a robot, but this may change as we dig in. The subject of this instalment of the Physics of Racing will therefore be to introduce the existing RARS physics model along with a long-range plan for enhancing the physics. I am very grateful to the authors for supplying RARS and I hope they will enjoy what I do with their work. The program is very easy to build, run, understand, and enhance. I encourage you to download it and follow along with me. However, my articles will be self-contained: you won’t need to build and run RARS to understand what I’m doing with it.

I have found that there is another independent effort afoot to enhance RARS. It’s called TORCS and can be found at http://torcs.free.fr. This includes *some* of the enhancements I intend to make, but its goals are like those of RARS rather than like mine. It looks very promising, but it has three features that make it unsuitable as a starting point for me:

- it's unfinished, whereas RARS is functional and established
- it's Linux-based. I don't have a Linux development environment, and it would take me too much time and money to build one up at present
- as usual, peeking (too much) at other work would spoil the fun for me

However, I will be keeping an eye on TORCS. It may turn out to be terrific!

My first approach to adapting RARS to a line-finding task will be to write a robot that learns the optimal line by making small modifications on each lap around the track, much as a human driver would do. This is a kind of *variational* approach, common in physics. The line-finding robot (LFR) will build an internal memory of its current line and everything it discovers about the track. Then, it will tweak the line, and, if the lap time goes down, continue to tweak in the same direction. Otherwise, it will discard the tweak and try another. At the point of diminishing returns, it will start tweaking another part of the line.

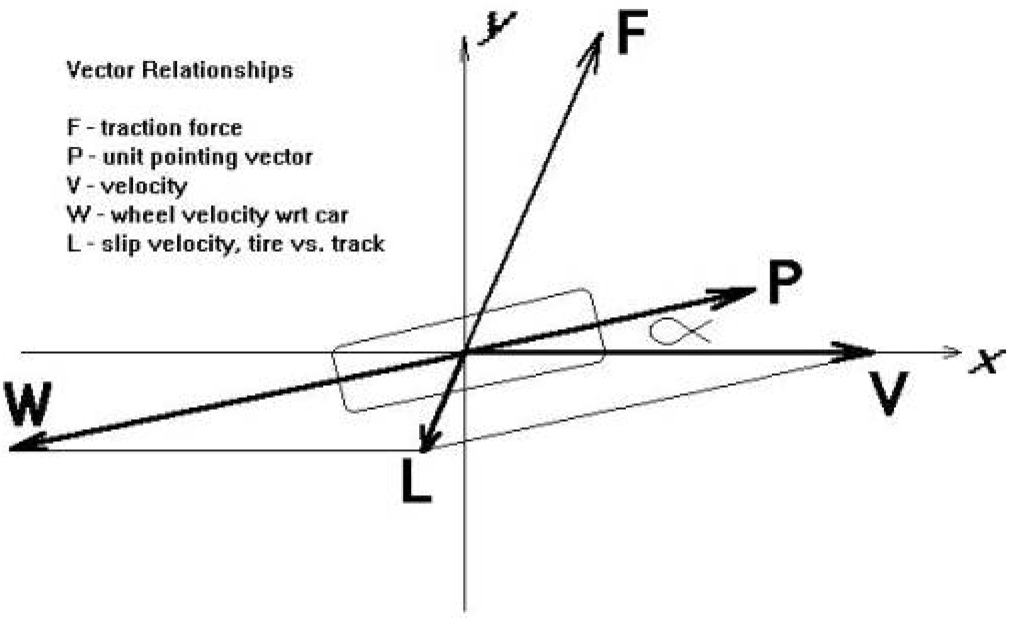
It's going to consume a lot of computing resources and not be competitive in the real-time setting of the old RARS. However, remember, with RARSEP, we are changing the goals. Also, this plan may take considerable time and span many articles. It may not work out at all. As usual, I am taking you along for the ride.

So, let's describe the current physics model. RARS' algorithm is devilishly simple, just the right compromise between physics rich enough to be convincing yet not so complicated that writing a robot is too challenging. Every time step, the simulation engine gives each robot a **situation** structure, and the robot responds with a command or **control** structure. The situation structure contains the current location and velocity of the car relative to the track, the walls, and the other cars. The control structure declares the desired **slip angle** - roughly representative of the steering-wheel angle - and the desired forward velocity - roughly representative of the throttle (positive) and brake (negative). The controls interact with the road through a tyre friction model, generating a force that accelerates the car. The force is limited by the power available from the engine, so, it is not always the case that *all* the force the tyres *could* deliver can be applied, since the engine may not be able to pump it out. So, the desired velocity may not be the achieved velocity.

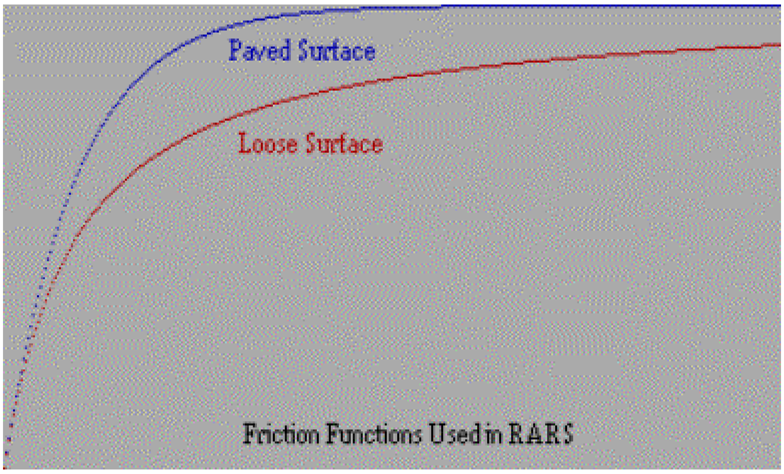
One reason that RARS is simple is that it is two-dimensional. In 2-D RARS, there are three, right-handed coordinate systems. First is the **ground**, a nearly inertial coordinate system fixed with respect to the road. Forces and accelerations are computed in this system, since it is inertial. Second is the **car** coordinate system. The **x-**axis of **car** points forward and the **y-**axis points to driver's left. Third and final is the **path** coordinate system, aligned with the car's velocity vector. The **tangential** component of any vector points along the **x**-axis of **path**, and the **normal** component of any vector points along the **y**-axis of **path**. **Car** aligns with **path** only when the car has no slip angle.

The following table, adapted from the program documentation, summarizes the physics model:

V	= car's velocity vector WRT (with respect to) ground
v	= ground speed = magnitude of **V**
P	= forward-pointing unit vector in the **car** system
alpha	= "slip angle" [command output from robot], which separates **P** and **V**. **Alpha** is positive when the car points to the left of **V**, as when power-sliding around a left-hand corner.
W	= velocity vector of tyre contact patch WRT **car**, always points backwards along x axis
vc	= "velocity commanded", [command output] forward in the **car** system; **W** = - **P** * **vc**
L	= **V** + **W** = **V** - **P** * **vc** = "slip vector", velocity of contact patch WRT ground
Lt	= **path**-tangential component of **L** = **v** – **vc** * cos(**alpha**)
Ln	= **path**-normal component of **L** = **v** - **vc** * sin(**alpha**)
l	= slip speed = magnitude of **L**
Q	= **L** / **l** = unit vector in the direction of **L**
mu(l)	= coefficient of friction, depending only on slip speed
F	= - **Q** * mass * **mu**(**l**) = force vector pushing the car, in the direction opposite to **L**
f	= mass * **mu**(**l**) = magnitude of **F**
Ft	= **path**-tangential component of **F** = **- f** * **Lt** / **l**
Fn	= **path**-normal component of **F** = **- f** * **Ln** / **l**
FtP	= projection of **Ft** in the **car** system = **Ft** * cos(**alpha**)
FnP	= projection of **Fn** in the **car** system = **Fn** * sin(**alpha**)
pwr	= engine power consumed = sum of force components along **P** limited by engine capacity = max(181hp, (**FtP** + **FnP**) * **vc**)



The friction function currently used is of the form **u(l) = FMAX * l / (K + l)** where **FMAX** and **K** are given constants.



To summarize the limitations of the current model:

- Track
  - Flat, fixed-width, no bumps

- Car
  - Point mass, no suspension

Planned enhancements:

- Track:
  - Elevation changes
  - Width variation
  - Camber, banking
  - Crown, profile
  - FIA berms
  - Bumps

- Car:
  - Four wheels
  - Discrete transmission, gear changes
  - Suspension: springs, dampers
  - Aerodynamics

As we progress, it may be helpful to keep these pages around. We will refer to them frequently.

The Physics of Racing, Part 17: "Slow-in, Fast-out!" or, Advanced Analysis of the Racing Line

Brian Beckman, PhD

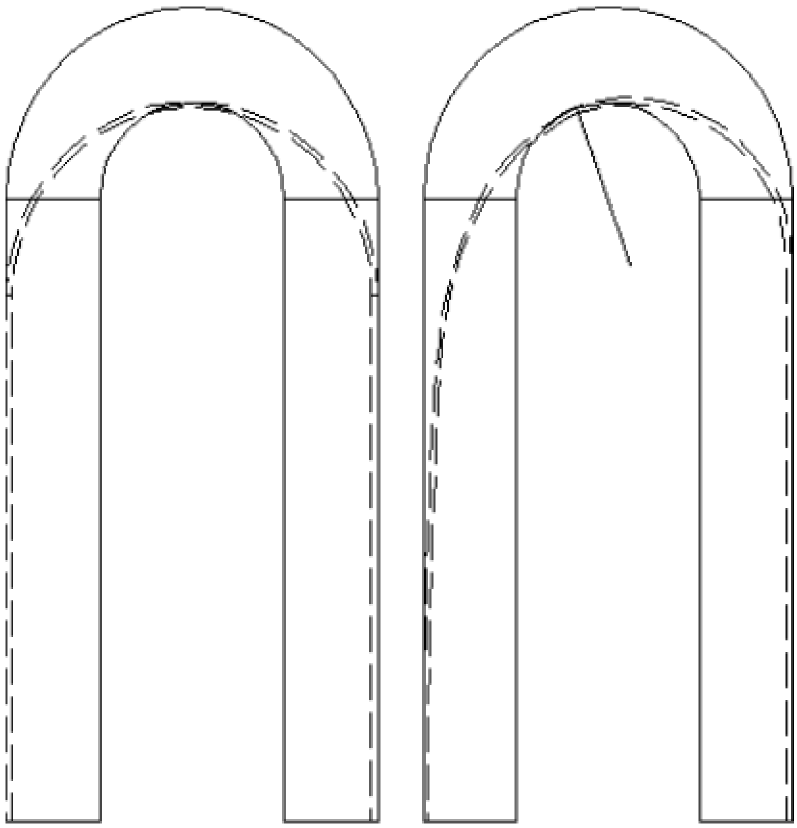
You may remember way back in part 5 that we did some simple calculations by hand to show that the classic racing line through a 90-degree right-hander is better than the either the line that hugs the inside or the line that hugs the outside of the corner. 'Better' means 'has lowest time.' The 'classic racing line' was, under the assumptions of that article, the widest possible inscribed line.

In this and the next instalment of *The Physics of Racing*, we raise the bar. Not only do we calculate the times for *all* lines through a corner, but we show a new *kind* of analysis for the exit, accounting for simultaneously accelerating and unwinding the steering wheel after the apex. This kind of analysis requires us to *search* for the lowest time because we cannot calculate it directly. We apply the approximation of the traction circle–subject of part 7–to stay within the capabilities of the car. We also model a more complex segment than in part 5, including an all-important exit chute where we take advantage of improved corner-exit speed. This style of analysis applies directly to computer simulation that we now have in progress in other continuing threads of *The Physics of Racing*.

The whole point of this analysis is to back up the old mantra: "slow-in, fast-out." We will find that the quickest way through the whole segment does *not* include the fastest line around the corner. Rather, **we get the lowest overall time by cornering more slowly so we can get back on the gas earlier**. It's always tempting to corner a little faster, but it frequently does not pay off in the context of the rest of the track.

This analysis is sufficiently long that it will take two instalments of this series. In this, the first instalment, we do exact calculations on a ***dummy line***, which is the actual line we will drive up to the apex, but just a reference line after the apex. In the next instalment, we improve on the dummy line by accelerating and unwinding, predicting the times for a line we would actually drive, but entailing some small inexactitude.

Let's first describe the track segment. Imagine an entry straight of 650 feet, connected to a 180-degree *left-hander* with outer radius 200 feet and inner radius 100 feet, connected to an exit chute of 650 feet. In the following sketch, we show the segment twice with different lines. The line on the left contains the widest possible inscribed cornering radius, and therefore the greatest possible cornering speed. The sketch on the right shows the line with the lowest overall time. Although its cornering speed is slower than in the line on the left, it includes a lengthy acceleration and unwinding phase on exit that more than makes up for it.



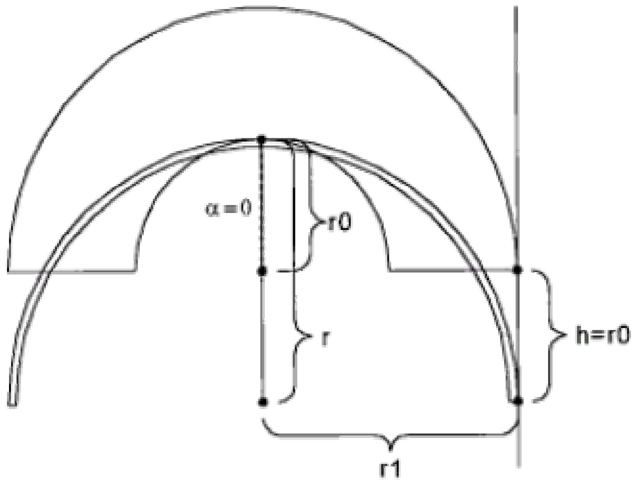
Line with
Fastest Cornering Speed

Line with
Lowest Overall Time

Note that *both* lines begin on the extreme right-hand side of the entry straight. Such will be a feature of every corner we analyse. Lines that begin elsewhere across the entry straight may be valid in scenarios like passing. However, we focus here on lines that are more obvious candidates for lowest times. Also, throughout, we ignore the width of the car, working with the 'bicycle line'. If we *were* including the width, ***w***, of the car, we would get the same final results on a track with outer radius of **200 + *w* / 2** feet and inner radius of **100 - *w* / 2** feet.

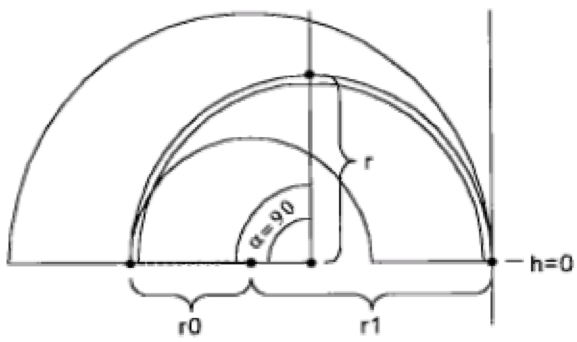
First, we compute exact times where we can on the course: the entry straight, the braking zone, and the corner up to the apex. To have a concrete baseline for comparison, we also do a 'suboptimal' exit computation–the dummy line–that includes completing the corner without unwinding and then running down the exit chute dead straight somewhere in the middle of the track. In the next instalment of *The Physics of Racing*, we compare the dummy line to the more sophisticated exit that includes simultaneously accelerating and unwinding to use up the entire width of the track in the exit chute.

Let us enter the segment in the right-hand chute at 100 mph = 146.667 fps (feet per second). We want the total times for a number of different cornering radii between two extremes. The largest extreme is a radius of 200 feet, which is the same as the radius of the outer margin of the track. It should be obvious that it is not possible to drive a circle with a radius greater than 200 feet and still stay on the track. This extreme is depicted in the following sketch:



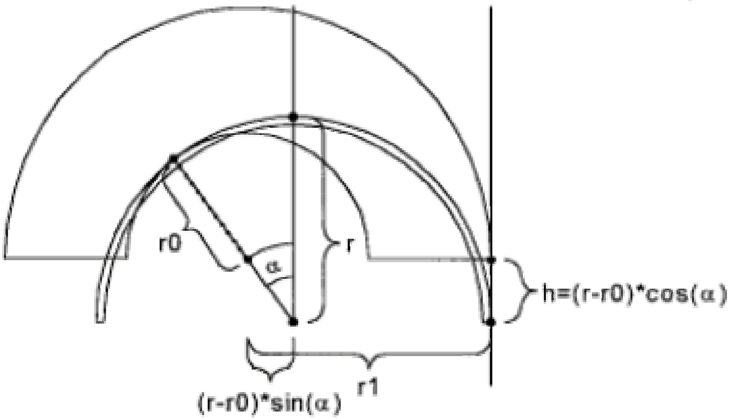
Extreme Case:
Widest Possible Line

We take the opportunity, here, to define a number of parameters that will serve throughout. First, let us call the radius of the outer edge of the track ***r*1**; this is obviously 200 feet, but, by giving it a symbolic name, we retain the option of changing its numeric value some other time. Likewise, let's call the radius of the inner circle ***r*0**, now 100 feet. Let's use the symbol ***r*** to denote the radius of the inscribed circle we intend to drive. In the extreme case of the widest possible line, ***r*** is the same as ***r*1**, namely, 200 feet. In the other extreme case, that of the tightest inscribed circle, ***r*** is 150 feet, as shown in the following sketch:



Extreme Case:
Tightest Possible Line

We're now ready to discuss the two remaining parameters you may have noticed: ***h*** and ***α*** (Greek letter alpha). Consider the following figure illustrating the general case:



General Case, Including the Intermediate Case:
Line with Lowest Overall Time

h indicates the point where we must be done with braking. More precisely, ***h*** is the distance of the turn-in point *below* the geometric start of the corner. Its value, by inspection, is **(*r* - *r*0) cos *α***. ***α*** is the angle past the geometric top where the inscribed circle-the driving line-apexes the inner edge of the track. We see two values for the horizontal distance between the centre of the inscribed circle and the centre of the inner edge, and those values are **(*r* - *r*0) sin *α*** and ***r*1 - *r***. Their equality allows us to solve for ***α***:

$$\alpha = \sin^{-1}\left(\frac{r1 - r}{r - r0}\right)$$

The following table shows numeric values of ***h*** and ***α*** for a number of inscribed radii (Note that if we varied ***r*0** and ***r*1** we would have a much larger 'book' of values to show. For now, we'll just vary ***r***.):

Inscribed Corner Radius (ft)	α (deg)	*h* (ft)
150	90.00	0.00
151	73.90	14.14
152	67.38	20.00
153	62.47	24.49
154	58.41	28.28
155	54.90	31.62
160	41.81	44.72
165	32.58	54.77
170	25.38	63.25
175	19.47	70.71
180	14.48	77.46
185	10.16	83.67
190	6.38	89.44
195	3.02	94.87
200	0.00	100.00

There are a couple of interesting things to notice about these numbers. First, they match up with the visually obvious values of $\boldsymbol{h} = 0$, $\boldsymbol{\alpha} = 90$ and $\boldsymbol{h} = 100$, $\boldsymbol{\alpha} = 0$ when $\boldsymbol{r} = 150$, $\boldsymbol{r} = 200$ respectively. This is a good check that we haven't made a mistake. Secondly, $\boldsymbol{\alpha}$ changes very rapidly with corner radius, and this fact has *major* ramifications on driving line. ***By driving a line just one foot larger than the minimum, one is able to apex more than fifteen degrees later!***

With these data, we're now equipped to compute all the times up to the apex and beyond. First, let's compute the speed in the corner by assuming that our car can corner at $\mathbf{1}\boldsymbol{g} = \mathbf{32.1\ ft / s^2} = \boldsymbol{v}^2 / \boldsymbol{r}$, giving us $\boldsymbol{v} = \sqrt{\boldsymbol{gr}}$. We express all speeds in miles per hour, but other lengths in feet. We won't take the time and space to write out all the conversions explicitly, but just remind ourselves once and for all that there are 22 feet per second for every 15 miles per hour.

Now that we have the maximum cornering speed, we can compute how much braking distance we need to get down to that speed from 100 mph. Let's assume that our car can brake at 1*g* also. We know that braking causes us to lose a little velocity for each little increment of time. Precisely, $\boldsymbol{dv / dt = g}$. However, we need to understand how the velocity changes with distance, not with time. Recall that $\boldsymbol{dx / dt = v}$, $\boldsymbol{dt = dx / v}$, so we get $\boldsymbol{dx = vdv / g}$. Those who remember differential and integral calculus will immediately see that $\boldsymbol{\Delta x} = \frac{1}{2\boldsymbol{g}}\left(\boldsymbol{v}_1^2 - \boldsymbol{v}_2^2\right)$ is the required formula for braking distance. In any event, the braking distance goes as the square of the speed, that is, like the kinetic energy, and that's intuitive. However, there's a factor of two in the numerator that's easy to miss (the origin of this factor is in the calculus, where we compute limit expressions like $(\boldsymbol{v} + \boldsymbol{dv})^2 \approx \boldsymbol{v}^2 + 2\boldsymbol{vdv}$).

We next subtract the braking distance from the entry straight, and also subtract $\boldsymbol{h}$, to give us the distance in which we can go at 100 mph, top speed, before the braking zone.

Now, we need the time spent braking, and that's easy: $\boldsymbol{\Delta t} = \boldsymbol{\Delta v} / \boldsymbol{g}$. All the other times are easy to compute, so here are the times for a variety of cornering lines *up to* the apices (or apexes for those who aren't Latin majors):

Inscribed Corner Radius (ft)	Cornering speed @1g in mph	Braking Distance (ft) @1g from 100 mph	Straight Distance (ft) prior to braking	Time (sec) in straight @ 100 mph prior to braking	Time (sec) in braking zone	Time (sec) in corner prior to apex	Total time (sec) up to the apex
150	47.24	261.11	388.89	2.652	2.418	6.802	11.872
152	47.55	260.11	369.89	2.522	2.404	5.987	10.912
154	47.86	259.11	362.60	2.472	2.390	5.682	10.544
155	48.02	258.61	359.77	2.453	2.382	5.566	10.401
160	48.79	256.11	349.17	2.381	2.347	5.144	9.872
170	50.29	251.11	335.64	2.288	2.278	4.641	9.208
180	51.75	246.11	326.43	2.226	2.212	4.325	8.762
190	53.16	241.11	319.45	2.178	2.147	4.099	8.424
200	54.55	236.11	313.89	2.140	2.083	3.927	8.150

At first glance, it appears that the widest line is a *huge* winner, but we must realize that these times include only driving up to the apex, and that is far earlier on the widest line, where $\boldsymbol{\alpha} = 0$. Suppose we continued driving all the way around the corner at constant speed and then accelerated out the exit chute at $0.5g$? This is the dummy line. We won't really drive this line after the apex, but discuss it nonetheless to provide a reference time. It's very easy to compute and provides a foundational intuition for the more advanced exit computation to follow in the next instalment:

Inscribed Corner Radius (ft)	Total time (sec) up to the apex	Time (sec) in corner after apex	Time for entrance and complete corner	Exit speed from chute (mph) @ g/2 accel	Time in exit chute (sec)	Combined segment time	Combined post-apex time and exit-chute time
150	11.872	0.000	11.872	109.091	5.670	17.541	5.670
152	10.912	0.860	11.773	107.857	5.528	17.301	6.388
154	10.544	1.209	11.754	107.422	5.460	17.213	6.669
155	10.401	1.348	11.750	107.260	5.430	17.180	6.779
160	9.872	1.881	11.753	106.697	5.308	17.061	7.189
170	9.208	2.600	11.808	106.101	5.116	16.924	7.716
180	8.762	3.126	11.888	105.806	4.955	16.844	8.082
190	8.424	3.556	11.980	105.666	4.813	16.792	8.369
200	8.150	3.927	12.077	105.627	4.682	16.760	8.609

So, we see that, driving the dummy line, the widest line yields the *slowest* time from the entrance up through the complete semicircle, but the quickest *overall* time when the exit chute is included. The widest line has lower (better) times than the tightest line in

- the entry straight by about half a second, because ***h*** is large and the entry straight is shorter for wider circles
- in the braking zone by about three tenths because the cornering speed is higher and less braking is needed
- and in the exit chute by almost a second, again because is ***h*** large and the exit chute is thereby shorter

The widest line has higher (worse) times by about a second in the circle itself because the wider circle is also longer. When the balances are all added up, the widest line is about eight tenths quicker than the tightest line, but it's ***all because of the effects of the corner on the straights before and after***.

Recall once again that the dummy line is not a line we would actually drive after the apex. But, with that as a framework, we're in position to introduce the next improvement. Everything we do from here on improves just the post-apex portion of the corner and the exit chute. We will actually drive the dummy line up to the apex. So, from this point on, we need only look at the last column in the table above, where we are shocked to see that there are almost three seconds' spread from the slowest to the quickest way out. A good deal of this ought to be available for improvement by accelerating and unwinding.

The Physics of Racing, Part 18: "Slow In, Fast Out!" or, Advanced Racing Line, Continued

Brian Beckman, PhD

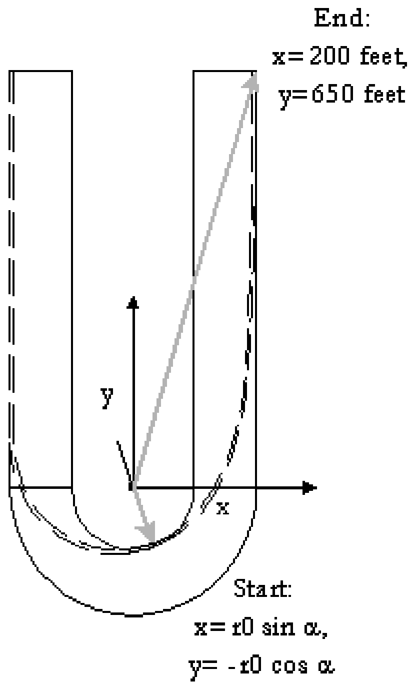
In the previous instalment, we did exact calculations for a dummy line down a 650-foot entry straight, a 180-degree left-hander, and a 650-foot exit chute. Cornering radii vary from 150 feet to 200 feet, and the track is 100 feet wide all the way around. This dummy line carries constant speed around the entire left-hander. We did those calculations to provide reference times to compare against this month's more sophisticated calculations, in which we unwind the steering wheel and accelerate at the same time. The baseline times for the dummy line over the whole course, as a function of cornering radius, are in the second-to-last column of the following table:

Inscribed Corner Radius (ft)	Total time (sec) up to the apex	Time (sec) in corner after apex	Time for entrance and complete corner	Exit speed from chute (mph) @ g/2 accel	Time in exit chute (sec)	Combined segment time	Combined post-apex time and exit-chute time
150	11.872	0.000	11.872	109.091	5.670	17.541	5.670
152	10.912	0.860	11.773	107.857	5.528	17.301	6.388
154	10.544	1.209	11.754	107.422	5.460	17.213	6.669
155	10.401	1.348	11.750	107.260	5.430	17.180	6.779
160	9.872	1.881	11.753	106.697	5.308	17.061	7.189
170	9.208	2.600	11.808	106.101	5.116	16.924	7.716
180	8.762	3.126	11.888	105.806	4.955	16.844	8.082
190	8.424	3.556	11.980	105.666	4.813	16.792	8.369
200	8.150	3.927	12.077	105.627	4.682	16.760	8.609

From this point on, we need only look at the last column. It's after the apex and down the exit chute where we look for improvement; we actually drive the dummy line up to the apex. Many readers will be screaming that we *could* try to get on the gas *before* the apex for even *more* improvement. Others will be screaming "trail brake!," that is, ease off the brakes at the same time as winding the steering wheel at turn in (thanks to reader Marc Sibilia for pointing this out to me). We leave those refinements to later articles.

The approach in this article is to find a line by building it up, step-by-step, honouring the traction circle and the sides of the track. This is one of the techniques we can use in computer simulations, so we get to kill two birds with one stone: previewing simulation and analysing a particular driving line. For convenience, we need a Cartesian coordinate system, that is, a square grid. Let's turn the track around 180 degrees for this purpose, and put the centre of the coordinate system at the centre of the corner. Since the inside edge of the track and the outside edge of the track are concentric semicircles, there is only one identifiable centre of the corner.

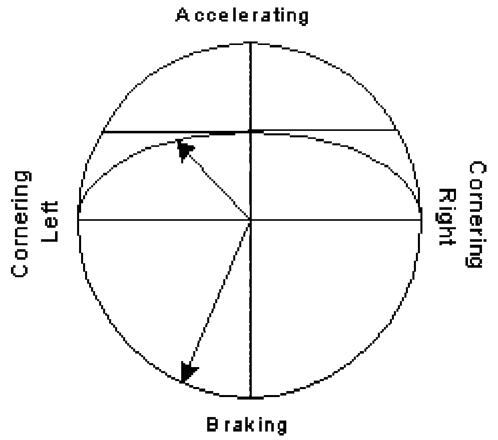
We'll work by measuring the position and heading of the centroid of the car with respect to this new coordinate system. We have a goal of arriving at the point $x = 200$, $y = 650$, measured in feet, in the least possible time, with a heading of as close to 90 degrees as we can get it, that is, heading straight down the track. We start at the apex, which measures from $x = r_0 \sin\alpha$, $y = r_1 \cos\alpha$. The following sketch illustrates:



I must note, at this point, if you haven't already noticed, this instalment of *The Physics of Racing* is going to be more concentrated and intense than previous instalments. I'm just going to blurt out facts without the usual explanations and walkthroughs. The reasons are (1) that we have a lot to get through in a little space and (2) that we assume that if you've been following the series this far, you've got the fortitude to work through it. So, *let's get it on!*

The initial heading is tangent to the inner edge of the track, that is, perpendicular to the line from the centre of the track's corner to the apex. Therefore, it has the angle α up from the horizontal x axis. We know the starting speed, $v0$, so we know its components in the x direction and in the y direction: $v0_x = v0\cos\alpha$, $v0_y = v0\sin\alpha$.

We perform the entire manoeuvre whilst never exceeding the limits of the traction circle. We set those limits as 1g cornering and braking and 0.5g accelerating, with smooth transitions all way around, as in the following sketch (the horizontal cap shows a way of accounting for engine limitations with *non*-smooth transitions, which will allow us to accelerate harder with the wheel still turned but probably scare us in the seat. Also, we note that 0.5g is a plausible, if only approximate, number for acceleration. We leave it to the reader to show that 0.5g in the quarter mile results in a realistic 13-second elapsed time, if at an unrealistic speed of 150 mph):



The Traction Circle

In each step of the calculation, we keep track of the following information:

- the time, t
- the current position, $x(t)$, $y(t)$, which we check to make sure we're still on the track $(x < 200)$ and to see whether we're done $(y \geq 650)$
- the current velocity, $v_x(t)$, $v_y(t)$, which we use to update the current position: $x(t+\Delta t) = x(t) + v(t) * \Delta t$, and likewise for y
- the tangential and radial acceleration, $a_t(t)$, $a_r(t)$, that is, tangential and radial to the bit of racing line at each instant (the *instantaneous* line), which we check to make sure that we're not cornering over the limit and that we're not exceeding the capacity of the engine, i.e., that $\sqrt{a_t^2 + a_r^2}$ is inside the traction envelope
- the acceleration in the x and y directions, $a_x(t)$, $a_y(t)$, which we use to update the current velocity: $v_x(t+\Delta t) = v_x(t) + a_x(t) * \Delta t$, and likewise for v_y

We drive the whole simulation by feeding on the throttle linearly with time over a time span called k and by simultaneously increasing the instantaneous radius of the driving line over a potentially different time span called k_{unwind}. Feeding on the throttle allows us to increase the tangential acceleration, a_t at each time step, and unwinding allows us to *decrease* the radial acceleration, a_r so we can stay within the traction circle. Since we'll still have centripetal traction available after the throttle is buried full on, we ought to be able to unwind more slowly, enabling us to stay on the track, but use it all up. In other words, we ought to look for solutions wherein k_{unwind} is larger than k, perhaps by twice.

Let's look at the first few rows of this simulation in a spreadsheet and delve into the formulas more deeply:

1	2	3	4	5	6	7	8	9	10	11	12
t	$a(t)$ (tangential, fpsps)	v^2/r (radial, fpsps)	$a(t)$ (radial, fpsps)	$r(t)$ (feet)	$a_x(t)$ (fpsps)	$a_y(t)$ (fpsps)	$x(t)$ (feet)	$y(t)$ (feet)	$v_x(t)$ (mph)	$v_y(t)$ (mph)	v (mph)
0.00	0.00	32.00	32.00	160.00	-21.33	23.85	66.67	-74.54	36.36	32.52	48.79
0.20	1.28	31.90	30.27	169.92	-21.20	21.64	76.80	-64.41	33.46	35.66	48.90
0.40	2.56	31.59	28.54	182.30	-20.76	19.75	86.09	-53.42	30.59	38.51	49.18
0.60	3.84	31.06	26.81	197.64	-20.06	18.19	94.54	-41.64	27.79	41.12	49.63
0.80	5.12	30.32	25.08	216.59	-19.17	16.96	102.20	-29.13	25.10	43.54	50.25
0.90	5.76	29.85	24.22	227.68	-18.67	16.47	105.74	-22.62	23.80	44.69	50.63
1.00	6.40	29.33	23.35	240.01	-18.13	16.05	109.09	-15.94	22.53	45.80	51.04

[column 1]: increments Δt by each row; we actually computed with $\Delta t = 0.05$ sec and display here every fourth actual row; this is an independent column, meaning that it does not depend on data from any other column.

[column 2]: tangential acceleration, $a_t(t) = \frac{g}{2}\min\left(1, \frac{t}{k}\right)$, accounting for squeezing on the throttle up to g / 2; depends only on column 1.

[column 3]: maximal radial acceleration, $v(t)^2 / r(t) = \sqrt{g^2 - 4a_t(t)^2}$, accounting for the traction circle; more precisely, for the upper half of the circle treated as a flattened (*oblate*) ellipse with height g / 2; depends only on column 2.

[column 4]: radial $a_r(t) = \max\left(0, \min\left(\frac{v(t)^2}{r(t)}, g\left(1 - \frac{t}{k_{unwind}}\right)\right)\right)$, accounting for unwinding the steering wheel; in steps from the inner parentheses outwards: $g(1 - t/k_{unwind})$ slowly decreases from g as time increases from 0, but, it is never allowed to exceed v^2 / r, by the **min** expression, as mandated by the traction circle, and then, never allowed to be negative, by the **max** expression, because we don't want to start turning back toward the entry straight; depends on columns 1 and 3.

[column 5]: $r(t) = v(t)^2 / a_r(t)$; just for amusement, it's interesting to calculate the instantaneous radius of a circle we could be driving if we were not accelerating tangentially; depends on columns 4 and 12, but no other columns depend on this.

[column 6]: $a_x(t) = \min\left(0, \frac{a_t v_x - a_r v_y}{v}\right)$, this just selects out the x components of both the radial and tangential accelerations, but makes sure that we never turn the wheel so much that we start going to the left. Note that the radial acceleration *always* tries to pull the car to the left, hence the minus sign (*centripetal*: see part 4 of *The Physics of Racing*); depends on columns 2, 4, 10, 11, and 12.

[column 7]: $a_y(t) = \min\left(0, \dfrac{a_t v_y + a_r v_x}{v}\right)$, selecting the y components, this time always pointing down the track, the way we want to go; depends on columns 2, 4, 10, 11, and 12.

[column 8]: $x(t) = x(t - \Delta t) + v_x(t)\Delta t$, just update the x coordinate by the velocity from the prior time step; depends on columns 8 (the prior row of itself) and 10.

[column 9]: $y(t) = y(t - \Delta t) + v_y(t)\Delta t$, do likewise for the y coordinate; depends on columns 9 (prior row) and 11.

[column 10]: $v_x(t) = \max(0, v_x(t - \Delta t) + a_x(t - \Delta t)\Delta t)$, for updating the x component of the velocity (but don't let it go negative, checking yet again, and, yes, this is a *hack*); depends on columns 10 (prior row) and 6.

[column 11]: $v_y(t) = v_y(t - \Delta t) + a_y(t - \Delta t)\Delta t$, likewise for the y coordinate of the velocity; depends on columns 11 and 7.

[column 12]: finally, $v = \sqrt{v_x(t)^2 + v_y(t)^2}$, depends on columns 10 and 11.

I've packed all this in an Excel spreadsheet. The spreadsheet should be in the download package for readers who acquired this document electronically.

Enough talk! Let's *drive!* Driving means playing with the values of r, k, and k_{unwind}, and possibly even Δt, to find the lowest overall time at which columns 8 and 9 show 200 or less and 650 or more, respectively. In general, "playing with" should be a sophisticated process involving hill climbing, genetic search, simulated annealing, and other fancy strategies for finding the very best values. In a computer simulation, we'd do that. However, we can do a reasonable job, for the sake of demonstration, by just tweaking the numbers by hand in the spreadsheet.

I have to admit that as I did so, I got kinaesthetic feelings as if I where actually driving. When I 'ran off the track,' that is, picked numbers that gave me $x > 200$, I gritted my teeth and blushed. When I was still unwinding at the end, I got that panicky feeling of understeer, knowing that I wasn't going to stay on after the end of the segment, and so on.

The best values I found by hand are shown in the following table at $r = 167.5$, $k = 3.25$, and $k_{\text{unwind}} = 7.22$. That means that we take 3.25 seconds to bury the gas and 7.22 seconds to unwind the wheel. There are solutions with lower segment times, but, since we're still unwinding long after the segment is done, I reject these solutions as assuming too much about what's going on after our segment is done. With more track to work with, however, *we can find lots more time*. In fact, it's a slightly surprising fact that by taking 9 seconds to unwind at $r = 167.5$, $k = 3.25$, we lose hardly any time and stay 15 feet inside the outer edge. There is quite a bit of territory to investigate even in this simple model.

r	***k***	k_{unwind}	**Best time Found**	**Dummy Time**	**Dummy-Best**	**Best Total Time Found**
155	1.500	2.000	6.500	6.779	0.279	16.901
160	2.500	3.700	6.875	7.189	0.314	16.747
165	3.000	5.950	7.050	7.482	0.432	16.550
167.5	3.250	7.22	7.120	7.605	0.485	16.466
170	3.500	8.550	7.225	7.716	0.491	16.433
175	4.000	11.170	7.400	7.912	0.512	16.367
180	4.500	13.330	7.575	8.082	0.507	16.337
185	5.000	30.000	7.700	8.233	0.533	16.282

Since the best dummy time, with the widest possible circle, is 16.760, and the best time I found here was 16.466, **the improvement by unwinding and accelerating simultaneously is 0.294 seconds**. This is very significant. If the exit straight were longer, the improvement would be even more dramatic since it would continue to accumulate time down the straight.

Note that this does *not* involve changing the entry to the corner other than by slowing down! There is no trail braking or lifting-while-turning or other risk-taking going on at corner entry. There is a very important driving lesson, here: to go faster, it is not necessary to take risks on corner entry. It is, in fact, ***both safer and faster just to slow down on the entry***. The improved exit will follow naturally from the combination of looking far ahead and of being smooth. And that's not even fair!

There is no guarantee that this is the best possible improvement in the model. I found these numbers by 'seat-of-the-pants' tweaking. A more systematic or algorithmic search would very likely find better ones. In other words, I was able to find almost three tenths by just driving a better line without trying very hard at all. There is another driving lesson, here: ***just driving a better line gives better times time without changing the driver's margin for error***, that is, without getting deeper into the *g* limits of the machine.

For the future, we can start taking more risks to get even more improvement. We can risk accelerating before the apex and we can risk deeper entry by trail braking, that is, easing off the brake and winding up the steering wheel at the same time. These manoeuvres do entail more driver risk since they are new opportunities for loss of car control.

Erratum: in part 17, I wrote "By driving a line just one foot larger than the minimum, one is able to apex more than fifteen degrees later!". I should have written

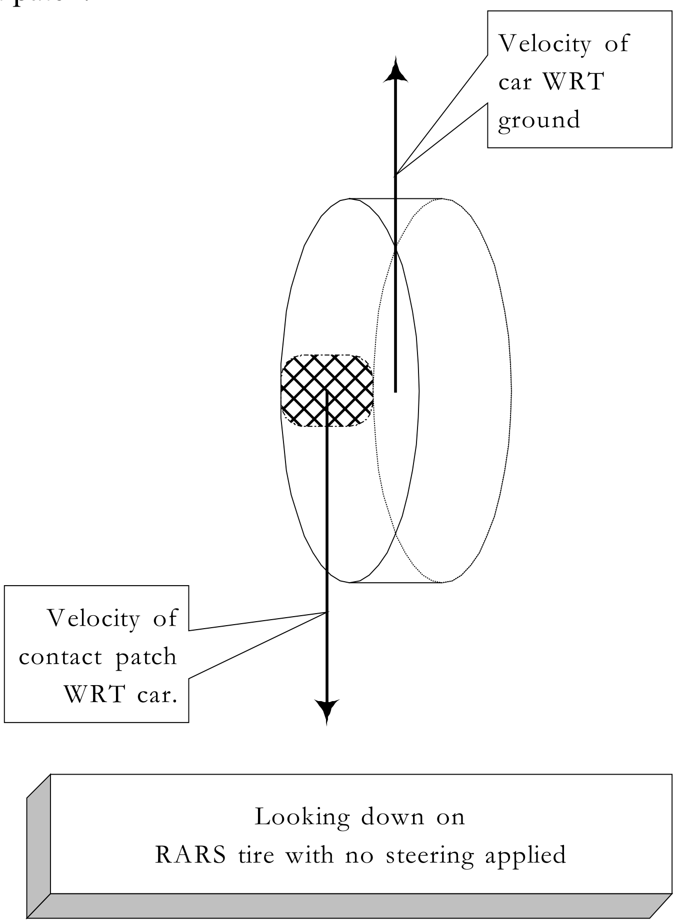
“…fifteen degrees *earlier*!” The point was that the tightest line does not apex until the geometric exit of the corner, and that’s *way too late*. The slip-of-the-pen occurred because one is so accustomed to talking about late apexing as preferable.

The Physics of Racing, Part 19: Space, Time, and Rubber

Brian Beckman, PhD

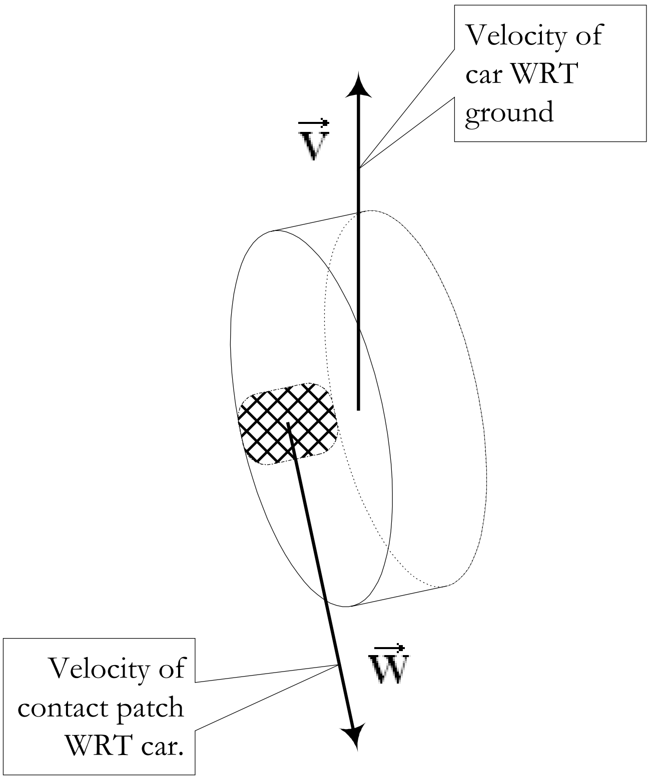
In part 16, we introduced RARS, the Robot Auto Racing Simulator. We talked about the clever and simple tyre-friction model in RARS and gave a terse presentation of its details in the big table in the article. Here, we'll explain in a little more detail why the model is cool.

First, consider RARS' idea of a tyre when there is no steering applied. In the following diagram, we look down on a RARS tyre from above, using "X-Ray Vision" to see the contact patch:



There are only two interesting quantities at this point: the velocity of the car with respect to the ground, and the velocity of the contact patch with respect to the car. If there is no power, braking, or cornering applied, then these two vectors are equal and opposite; in other words, the velocity of the contact patch *with respect to the ground* is zero. In general, if you have a velocity of some ***thing***, any *thing*, with respect to the **car** and you have the velocity of the car with respect to the **ground**, all you have to do to get the velocity of the thing with respect to the ground is add the vectors, and we show how to add vectors immediately below. You eliminate the middleman—the car—by so doing. This is *relativity*, not the Einstein kind, but the Galileo kind—hence the title of this article. In the Einstein kind of relativity, you correct the vector sum with some quantities depending on the speed of light, a constant, and this is not relevant for auto racing because the speeds are so low compared to the speed of light, which is about 670 million miles per hour.

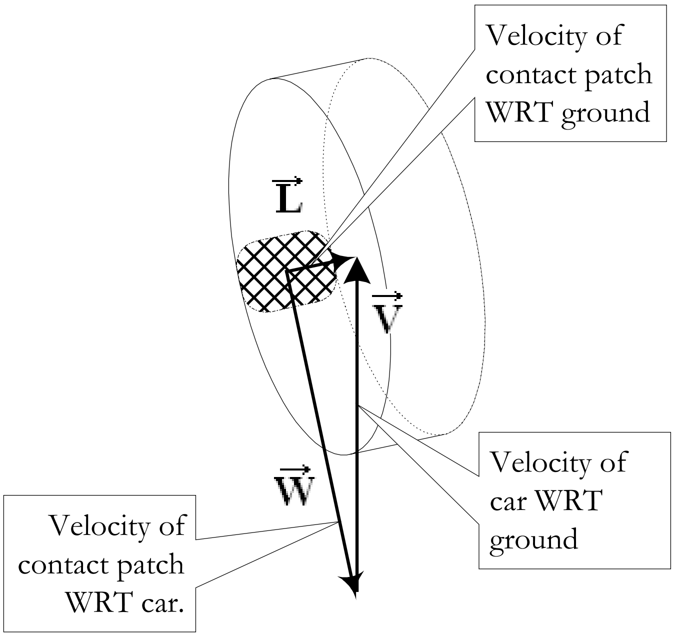
What happens when you apply a little steering input? Look at the next diagram.



Looking down on
RARS tire with slight left-hand lock

The velocity of the contact path with respect to the car, $\vec{W}$, gets a little angle—the slip angle or the grip angle, as I called it in part 10 of the *Physics of Racing*. $\vec{W}$ still points directly back along the plane of the tyre, and the velocity of the car with respect to the ground, $\vec{V}$, still heads forward, that is, *up* on the page, at least for an instant. To find out the velocity of the contact patch with respect to the ground, $\vec{L}$, we add the vectors $\vec{W}$ and $\vec{V}$, just as before, but now there's an intervening angle. Here's the procedure for adding the vectors:

Transport one of them, without changing its direction, so that its tail touches the head of the other one. It turns out that not all kinds of vectors can be transported like this, but velocity in this context is one of the kinds that we *can* freely transport. We'll transport $\vec{V}$ over to the head of $\vec{W}$, as in the following diagram:

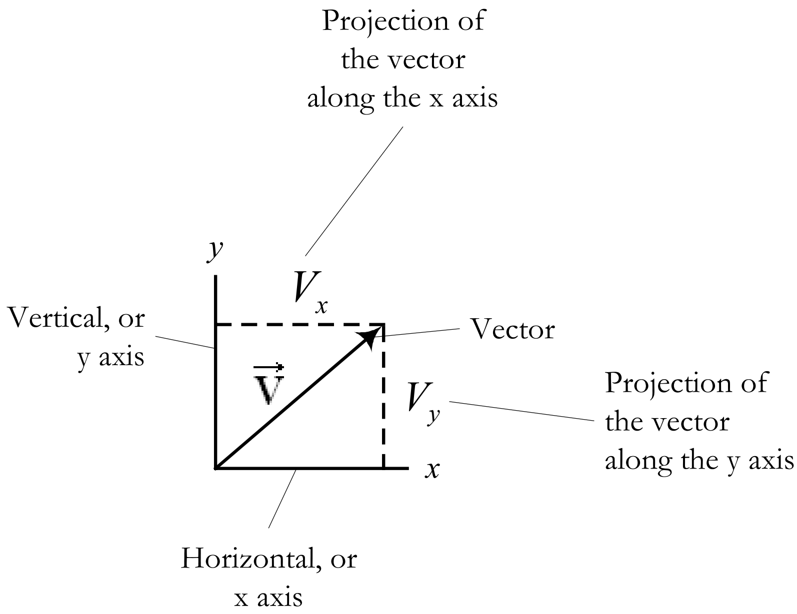


RARS tire with slight left-hand lock, showing contact patch velocity WRT ground

Draw a new vector from the tail of $\vec{W}$ to the head of $\vec{V}$ in its new location. This new, little vector is *defined as* the ***vector sum***, $\vec{L}$, drawn from the tail of one to the head of the other. Note it would have the same direction and length if it were drawn from the tail of $\vec{V}$ to the head of $\vec{W}$. Because of this fact, we can write the following equation:

$$\vec{L}=\vec{V}+\vec{W}=\vec{W}+\vec{V}$$

This procedure for adding vectors works even when the vectors are colinear, in which case the triangle is flat, the opposite corners coincide, and the vector sum is zero—the mathematically unique ***vector zero***. It also turns out that this procedure has a very simple equivalent in algebraic form. To do computations, we need to represent vectors with numbers. To do so, we measure the length of the projection of the vector against the axes of a coordinate frame, as in the following diagram:

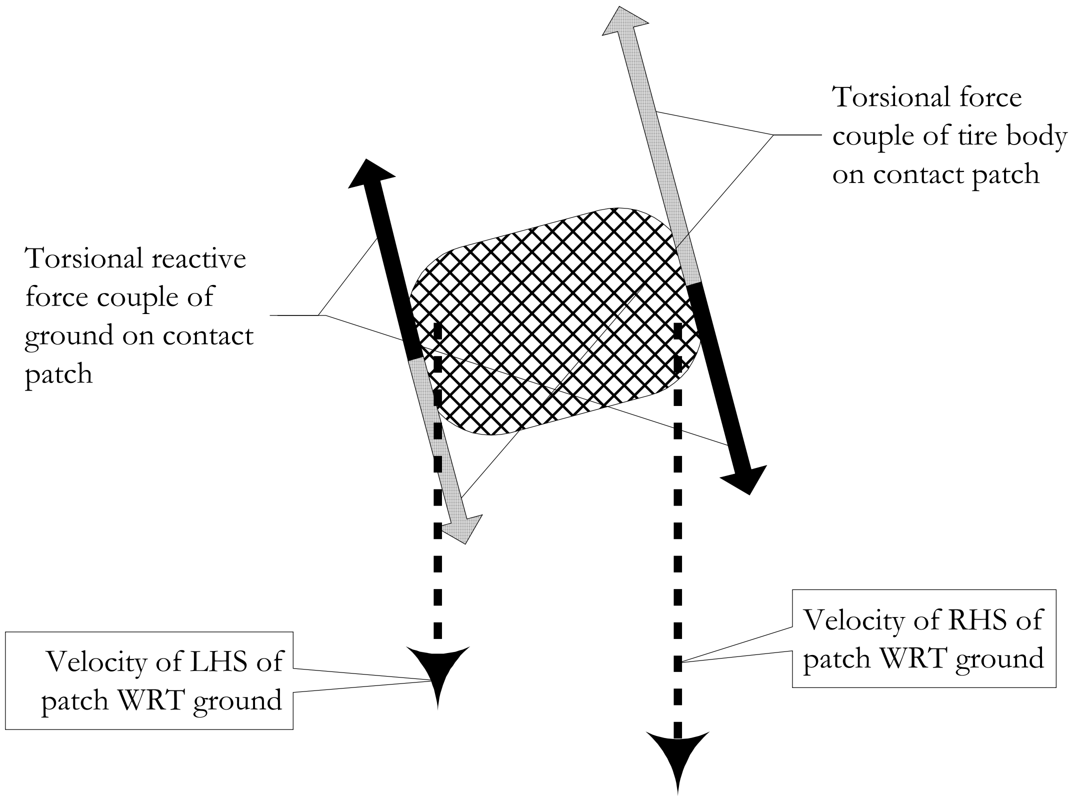


So, we get just what we need: numbers, also called ***components*** because they are the component parts that make up the perpendicular, independent projections of the vector. Numbers, in general, are also called *scalars* since they can be used to scale vectors. In general, we'll get three numbers for any vector in a 3D world, and two numbers in a 2D world. In either case, adding vectors is trivial. If we write $\vec{V} = V_x, V_y, V_z$ and $\vec{W} = W_x, W_y, W_z$,

then $\quad \vec{V} + \vec{W} = V_x + W_x + V_y + W_y + V_z + W_z$

Just add them up componentwise. Couldn't be easier. I'll leave it to you to show that the head-to-tail method is equivalent to the numerical add-'em-up-componentwise method.

Let's go back to the tyres. Here's what makes RARS' model so clever: It's undeniable that the contact patch moves with respect to the ground if we assume that it continues to move in the circumferential direction with respect to the wheel and the car. We can summarize all the complex motion of the contact patch in a single velocity $\vec{L}$ and we can approximate our friction model so that it depends only on the magnitude of $\vec{L}$. This is a simpler model than the one presented in part 10, but also potentially less accurate. Let's review that one briefly. Again, looking at the contact patch from above, as if by X-Ray vision:



In the diagram above, we're looking at a wheel steered slightly to the left of the direction of travel. Assuming that there's a little acceleration in addition to the steering, both sides of the contact patch move backwards with respect to the ground. The left-hand side (LHS) of the patch moves a little more slowly than the right-hand side (RHS) because the RHS crabs around the corner. The wheel, through the carcass of the tyre, twists the patch to the left, generating a force couple illustrated by the grey arrows. The ground resists, through friction, producing the right-twisting, restoring, force couple in black. Since the patch continues to twist leftward with respect to an inertial reference frame in a steady-state turn, the grey couple must be a little larger than the black one.

This model is not by any means complete, and it's already MUCH more complex than the RARS model, which does a decent job in practice. RARS computes just *one* quantity, the vector $\vec{\mathbf{L}}$, and accounts for all forces and torques on the tyre through that one variable. The advantages of the approach, when it comes to computer simulation, are

- very simple math, easy to code and debug
- very fast conversion from velocity to force
- one table lookup and one interpolation

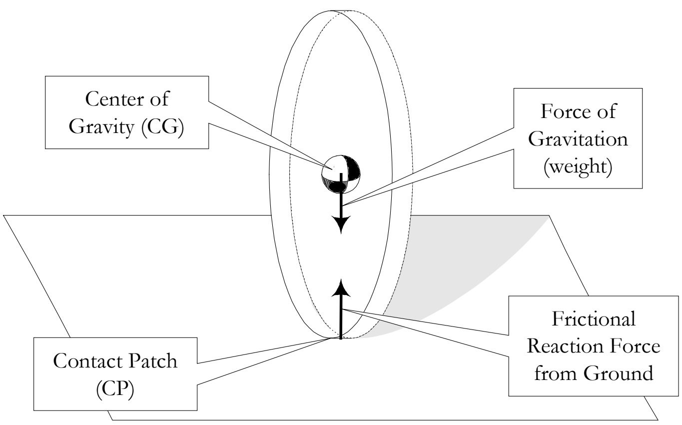
The limitations, of course, are that RARS cannot account for detailed tyre physics and important effects like suspension geometry and dynamics, so the whole scheme trades off accuracy for simplicity. However, as a quick-and-dirty approximation, it's remarkably effective.

The Physics of Racing, Part 20: Four-Point Statics

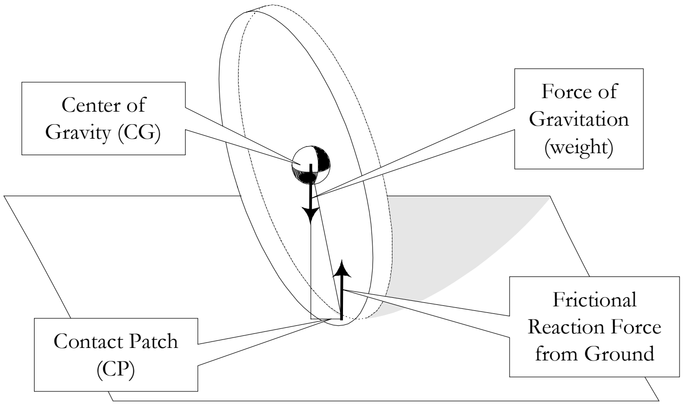
Brian Beckman, PhD

In this instalment, we analyse the stability of a single wheel, a bicycle, tricycle, and, finally, of a four-wheeled vehicle. In the offing, we introduce force moments, vector cross products, matrices and linear algebra, and some interesting facts about how the number of wheels on a vehicle relate to the number of dimensions of space and to the practice of weight jacking on a race car.

Consider a single bicycle wheel and tyre combination, all by itself, just standing on the ground. Call it a unicycle wheel. It almost immediately falls over. The reason is that its centre of gravity (CG) is above the ground, but its contact patch (CP) is ON the ground. Assuming that the CP doesn't slide, then the ground will resist any force put on it with an equal and opposite force. If the wheel begins to tip ever so slightly sideways, Earth's gravitation, pulling on the CG, and the reaction force, pushing mostly up and a little sideways on the CP, conspire to twist the wheel even more sideways down toward the ground. In other words, if it tips over just a little, it will have an overwhelming tendency to tip over ALL the way. The following figure shows the wheel precariously balanced on its CP:



The next figure shows the wheel just starting to tip over. One can easily see that the weight, pulling down on the CG, and the reaction force, pushing up on the CP, will quickly knock the wheel down to the ground. At any instance of time, the tendency for the wheel to fall over is measured by the ***moment*** of the forces about some arbitrary point. The moment of a force about a point is the magnitude of the force times the perpendicular distance of the force line from the point. We suggest this perpendicular distance in the following diagram with a small right triangle. Since the CP is not sliding, by assumption, it's fixed in inertial space and is an ideal candidate for a ***moment centre***: the point about which to compute force moments. There is also a small, sideways component to the ground's force on the CP, but we ignore that in the diagram.



More generally, the moment of a force vector can be thought of as a vector in its own right (at least in three dimensions, it can; the story is more complicated in four or more dimensions). This vector is the ***cross product*** of the ***moment arm*** and the force vector. The moment arm is a vector drawn from the moment centre to the point of application of a force. In the diagram above, the moment arm of the gravitational force is along the hypotenuse of the little triangle and points upwards. The cross product of a moment arm, **r**, and a force, **F**, is *defined to be* a certain vector that is perpendicular to both ***r*** and ***F***. There are lots of vectors perpendicular to both **r** and **F** if **r** and **F** are not collinear, and there are NO vectors perpendicular to both of them if they are collinear. In any event, we're looking for the particular one that satisfies some properties. Suppose **r** has components (*rx, ry, rz*) and **F** has components (*Fx, Fy, Fz*). Let the vector we're seeking be **T**. The conditions that **T** be perpendicular to **r** and to **F** can be written as follows (assuming you understand the much-simpler inner product or dot product of vectors—if not, take a search for "inner products" at http://www.britannica.com/ for one of many Internet sources):

$$\mathbf{T} \bullet \mathbf{r} = 0 = T_x r_x + T_y r_y + T_z r_z$$
$$\mathbf{T} \bullet \mathbf{F} = 0 = T_x F_x + T_y F_y + T_z F_z$$

It's easy to check that the following vector satisfies these two equations in three unknowns:

$$\mathbf{T} \overset{\Delta}{=} \mathbf{r} \times \mathbf{F} = (r_y F_z - r_z F_y, \quad r_z F_x - r_x F_y, \quad r_x F_y - r_y F_x)$$

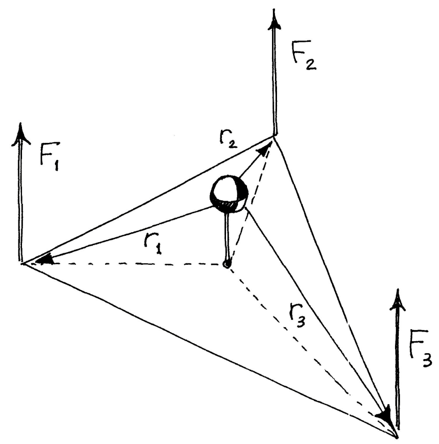
It's a little more subtle to check that the magnitude of **T**, written $\|\mathbf{T}\|$, is the magnitude of **r** times the magnitude of F times the **sin** of the angle between them in the plane they form. My favourite method is to do the calculation in that plane, where it's easy, then to assert that nothing in the result depends on the orientation of that plane in 3-space, so the answer must be the same after the plane is rotated into any arbitrary orientation. For the masochist, it's possible to prove by grinding through all the algebraic arithmetic that

$$\|\mathbf{T}\| = \|\mathbf{r}\|\|\mathbf{F}\|\sin\theta$$

Furthermore, the cross product we need satisfies the ***right-hand rule***, whereby **r**, rotated into **F** by a right-handed twist, as though by turning a right-handed screw or faucet handle, would advance the screw in the direction of **T**. The opposite product, $\mathbf{F} \times \mathbf{r}$, would have the opposite sign. There are many more interesting properties of the cross product, for which we refer you again to http://www.britannica.com/.

Let's go back to our unicycle wheel. Generically, a physical system is ***unstable*** if small inputs lead to large outputs, say, if ambient forces amplify little disturbances. The fact that our wheel falls over with just the slightest disturbance, almost by itself, indicates that the one-wheel system is unstable. By the way, a wide race tyre will not tip over by itself until it's pushed sideways far enough that the line of the gravitational force vector lies outside the edge of the tyre. At that point, the restoring force, pushing up on the edge of the tyre, can no longer counteract the tipping-over, twisting tendency of gravitation. A complete, ride-able unicycle is even *more* unstable than a unicycle wheel, because a rider must also keep himself from falling backwards or forwards by continuously adjusting pressure on the pedals. A ride-able bicycle does not suffer forward-backward instability since the CG is between the front CP and the rear CP. However, it does suffer left-right instability, and the rider must continuously adjust body weight and steering input—which generates sideways restoring force—to keep the bike from falling over. Going from one wheel to two wheels eliminates one form of instability. How about going to three wheels?

A tricycle is optimally stable. It will sit still without tipping over, and its stability in steady-state motion is almost exactly the same as its static—or standing-still—stability. It takes a large disturbance to knock over a tricycle. It will tend to come back down on its wheels even after becoming partially or completely airborne, so long as the CG stays within the bounds of the triangular outline of the CP's projected vertically on the ground (see the following figure):



The stability of a tricycle is reflected in the mathematics to solve for the normal forces on the contact patches when the tricycle is still or in steady motion in a straight line. Briefly, if the tricycle is NOT tipping over, the sum of the moments of the normal forces must vanish (the choice of the moment centre is arbitrary, but the CG is convenient because the weight has no moment about the CG). With reference to the preceding figure, we have the following vector equation:

$$\mathbf{0} = (\mathbf{r}_1 \times \mathbf{F}_1) + (\mathbf{r}_2 \times \mathbf{F}_2) + (\mathbf{r}_3 \times \mathbf{F}_3)$$

Since the forces are normal forces, they have only Z-components—that's what "normal" means when speaking of forces. The cross products are quite simple, then, and work out to be $\mathbf{r_i} \times \mathbf{F_i} = (r_{iy}F_{iz}, -r_{ix}F_{iz}, 0)$ for i = 1,2,3. Furthermore, we know that the forces must add up to the weight, **W**. We now go into the language of matrices and linear algebra to present the solution (you know the drill: go to Britannica if you're not comfortable), and, in the interest of space, we omit the intermediate arithmetic, which you may check on your own. We may write our equations as

$$\begin{pmatrix} r_{1y} & r_{2y} & r_{3y} \\ r_{1x} & r_{2x} & r_{3x} \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} F_{1z} \\ F_{2z} \\ F_{3z} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ W \end{pmatrix}$$

This set of equations has a solution if and only if the determinant of the square matrix is nonzero. This determinant is

$$d \equiv r_{1x}\left(r_{3y} - r_{2y}\right) + r_{2x}\left(r_{1y} - r_{3y}\right) + r_{3x}\left(r_{2y} - r_{1y}\right),$$

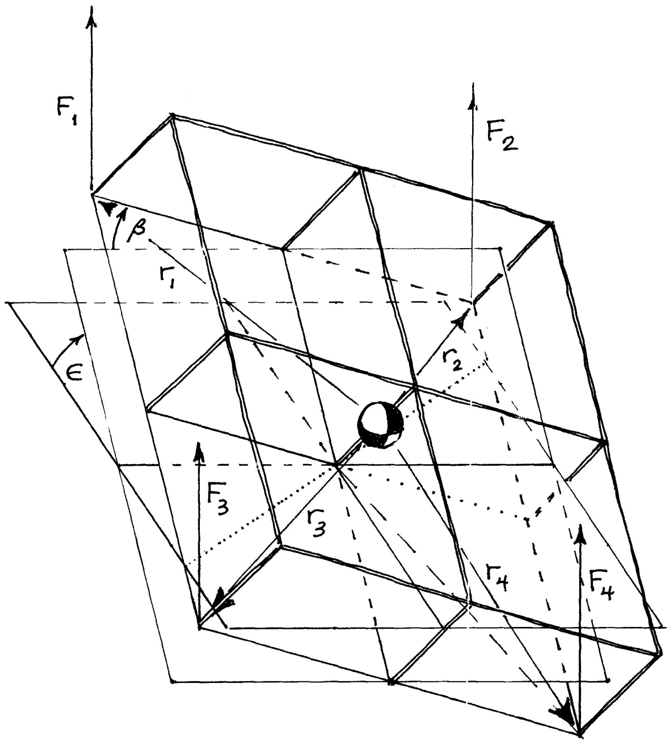
which will vanish if all three y components of the moment arms are equal, or, for that matter, if all three x components are equal. In other words, it will vanish if the three points of application of the forces are collinear, in which case we have three wheels in a line and we might as well have a bicycle as to stability. In any event, the solution turns out to be

$$\begin{pmatrix} F_{1z} \\ F_{2z} \\ F_{3z} \end{pmatrix} = \frac{W}{d} \begin{pmatrix} r_{2y}r_{3x} - r_{3y}r_{2x} \\ r_{3y}r_{1x} - r_{1y}r_{3x} \\ r_{1y}r_{2x} - r_{2y}r_{1x} \end{pmatrix}$$

Obviously, there is no solution if d, the determinant, vanishes. It is an interesting exercise to find out all the geometric circumstances in which one or more of these forces vanish or to catalogue all the possible ways in which the determinant can vanish. I will leave these exercises to the reader. Before leaving the tricycle, I'd like to assert without proof that the fundamental, geometric reason we can solve for the normal forces is that ANY three points define a plane. No matter how the tricycle is positioned on any (sufficiently horizontal) plane, all three wheels will touch the ground and three normal forces will be generated.

We now take a huge risk and generalize TWO aspects of the model at once. First, we go to four wheels. Second, we tilt the plane upward by a small angle $\boldsymbol{\varepsilon}$ and bank it by a small angle $\boldsymbol{\beta}$. Going to four wheels will cause our equations to break down because there is TOO much symmetry in the vehicle and blind application of linear algebra cannot derive, unambiguously, how the normal forces are to be apportioned among the wheels. Four points cannot lie on a plane unless they are exquisitely balanced there. We restore sanity by expressing the desired symmetry explicitly, and this makes for a bit of interesting math. Physically, in a four-wheeled vehicle with a suspension, it is very easy to load wheels preferentially by jacking the springs up or down. NASCAR crews are often furiously spinning wrenches above the rear wheel purchases in the pits, effectively jacking weight into or out of wheels to adjust handling. In a three-wheeled vehicle, weight jacking is not possible, to first order, that is, so long as the CG does not tip appreciably. Playing around a little with the spring heights on a tricycle will not affect the weight on each wheel.

Elevating and banking the plane further complicates the math by adding angle terms to the moment arms or to the forces, depending on point of view (we go with the latter). We turn on the fire hose, here, because it would take several instalments to go step-by-step. We just state the math and leave it to the adventurous reader to check it. First, a diagram:



Now, suppose the car has front track t_1, rear track t_2, distance w_1 from the CG to the front axle, distance w_2 from the CG to the rear axle, and height h_G of the CG off the ground. The lever arms, gathered in a matrix, are

$$\begin{pmatrix} w_1 & t_1/2 & -h_G \\ w_1 & -t_1/2 & -h_G \\ -w_2 & t_2/2 & -h_G \\ -w_2 & -t_2/2 & -h_G \end{pmatrix}$$

Let's abbreviate the four normal forces to (a,b,c,d) for (F_1,F_2,F_3,F_4). After elevation and banking, they are

$$\begin{pmatrix} a\sin\varepsilon & a\cos\varepsilon\sin\beta & a\cos\varepsilon\cos\beta \\ b\sin\varepsilon & b\cos\varepsilon\sin\beta & b\cos\varepsilon\cos\beta \\ c\sin\varepsilon & c\cos\varepsilon\sin\beta & c\cos\varepsilon\cos\beta \\ d\sin\varepsilon & d\cos\varepsilon\sin\beta & d\cos\varepsilon\cos\beta \end{pmatrix}$$

The four torques are too long to write out. We're really only interested in their sum, which works out to be the following 3-vector:

$$\begin{pmatrix} \cos\varepsilon\left(\tfrac{1}{2}p_2\cos\beta + p_3\sin\beta\right) \\ -p_1\cos\beta\cos\varepsilon - p_3\sin\varepsilon \\ p_1\cos\varepsilon\sin\beta - \tfrac{1}{2}p_2\sin\varepsilon \end{pmatrix} \quad \text{where} \quad \begin{aligned} p_1 &= \left[(a+b)w_1 - (c+d)w_2\right] \\ p_2 &= \left[(a-b)t_1 + (c-d)t_2\right] \\ p_3 &= h_G mg \end{aligned}$$

and where we have replaced $a + b + c + d$ with mg, the weight of the car, expressing force balance. We now have three equations in four unknowns, so we cannot solve without more information (in fact, the 4-matrix written out similarly to the tricycle example has a symbolically vanishing determinant—not good for physics, but it is the interesting mathematical point). Symmetry constraints are a typical way to add information, and a good symmetry is that the ratio of the two rear forces should equal the ratio of the two front forces, or $ad = bc$, expressing the circumstance that we have NOT jacked any weight into the car. As an intermediate step, we can solve the original set of equations for a in terms of b, c, and d, yielding

$$a = \left(bt_1 - ct_2 + dt_2 - 2h_G mg\tan(\beta)\right)\big/t_1$$

We can go one more step by solving for b in term of c and d by setting the torque in the PITCH, or y, direction to zero, yielding

$$b = \left[(c-d)t_2 w_1 + (c+d)t_1 w_2 - h_G mg\left(2w_1\tan\beta - t_1\sec\beta\tan\varepsilon\right)\right]\big/2t_1 w_1$$

Let's make a few simplifying definitions:

$$\begin{aligned} \zeta_1 &= h_G mg\left(2w_1\tan\beta - t_1\sec\beta\tan\varepsilon\right) \\ \zeta_2 &= h_G mg\left(2w_1\tan\beta + t_1\sec\beta\tan\varepsilon\right) \\ \xi &= t_1 w_2 + t_2 w_1 \end{aligned}$$

Writing out our symmetry equation and substituting the solutions for a and b above, we get (after a distressing amount of grungy grinding)

$$\left(c^2 - d^2\right)\xi + c\zeta_1 + d\zeta_2 = 0$$

The good ol' high-school formula for the solution of quadratic equations gives

$$\begin{aligned} c &= \left[\zeta_1 \pm \sqrt{\zeta_1^2 + 4d\xi\left(d\xi - \zeta_2\right)}\right]\big/2\xi \\ d &= \left[\zeta_2 \pm \sqrt{\zeta_2^2 + 4c\xi\left(c\xi + \zeta_1\right)}\right]\big/2\xi \end{aligned}$$

We’ve gone on long enough in this article. We’ll leave it to a later instalment or to the reader to work with some numerical values and plots.

The Physics of Racing, Part 21: The Magic Formula: Longitudinal Version

Brian Beckman, PhD

Driving a car is a classic problem in ***control***. Here, we mean *control* in the technical sense of *control theory*, an established branch of engineering science (once again, I find www.Britannica.com to have a very nice, brush-up article on that term). In a more-or-less continuous fashion, the driver compares desired direction, speed, and acceleration with actual direction, speed, and acceleration. The driver uses visual input to sense actual direction and speed; and uses visceral, inertial feedback—the butt sensor—for actual acceleration. When the actual differs too much from the desired, the driver applies throttle, brake, steering, and gear selection to change the actual. These inputs cause the tyres to react with the ground, which pushes back against the tyres, and through the suspension, pushes the body of the car and driver. Drivers in high-speed circumstances can also generate desired aerodynamic forces, as in slipstreaming, in the "slingshot pass," and in the Earnhardt TIP manoeuvre, where the driver "takes the air off" the spoiler of the car in front of him.

Tyres generate forces by sticking and sliding and everything in between. They transmit these forces to the wheels by elastic deformation. The elastic deformation is extremely complex and theoretical computation requires numerical solution of finite-element equations. However, despite fierce trade secrecy, industry and academia have reached apparent consensus in recent years on a formula that summarizes experimental and theoretical data. This so-called ***magic formula*** is not a solution to equations of motion—a solution in such a form is not feasible. It's just a convenient fitting of commonplace mathematical functions to data. It allows one to compute forces at a higher precision than something like RARS (see parts 16 and 19 of the Physics of Racing [*PhOR*]), but without integrating equations. Therefore, forces can be computed within a reasonable time, say in a real-time simulation program.

To understand the magic formula, we need first to define its inputs, which include ***slip***. Slip is an indirect measure of the fraction of the contact patch that is sticking. It is frequently asserted in the literature that a tyre with no slip at all cannot create forces. It has taken me a very long time to accept this assertion. Why can I steer a tin-toy car with metal tyres on a hard surface like Formica? If there is any slip in such tyres, it is microscopic, yet there are sufficient forces to brake and steer, even if just a little. I finally caved in when I realized that the forces are minute, also. If there is any friction between the tyre and the surface, there MUST be slip, as it is defined below. Though to a very small degree, the Formica and the tiny contact patches of the tin tyres actually twist and stretch each other. The only way to eliminate slip completely is to eliminate GRIP completely. Any grip, and you will have slip.

There are two, slightly different flavours of the magic formula. The ***longitudinal*** one is the subject of this entire instalment of PhOR, and we cover the ***lateral*** one in the next instalment. Longitudinal slip is along the mean plane of the wheel and might also be called *circumferential* or *tangential*. It creates braking and accelerating forces. Lateral slip is our old friend grip angle [PhOR-10], and it generates cornering forces.

We write longitudinal slip as σ. It's defined by the *actual* angular velocity, ω, of a wheel plus tyre under braking or acceleration, compared to the corresponding angular velocity of the same wheel plus tyre when rolling freely. We write the free-rolling angular velocity as $\omega_0 = V/R_e$, where V is the current, instantaneous velocity of the hub centreline of the wheel with respect to the ground, and R_e is the ***effective radius***, a constant defined below. Since the dimensions of V are length/time, and the dimensions of any radius are length, the ratio, $\omega_0 = V/R_e$, has dimensions of inverse time. In fact, it should be viewed as measuring radians per unit time, radians being the natural, dimensionless measure of angular rotation. There are 2π radians in one rotation or one circumference of a circle, just as the length of the circumference is 2π times the radius.

Let's begin the discussion of longitudinal slip with a question. Consider a wheel-tyre combination with 13-inch radius or 26-inch diameter, say a 255-50/16 tyre on a 16-inch rim. The "50" in the tyre specification is the ratio of the sidewall height to the tread width, which is also written into the specification as 255, millimetres understood. We get a sidewall height of 50 percent of 255 mm, which is 5.02 inch. Therefore, the total, ***unloaded radius***, half of the tread-to-tread diameter, is about 5+16/2 = 13 Inch. Now consider a rigid tyre of the same radius, made, say, of steel or of wood with an iron tread like old Western wagon wheels. The question is whether, given a certain constant hub velocity, pneumatic tyres spin faster than, slower than, or at the same speed as equivalent rigid tyres?

At first glance, one might say, "Well, faster, obviously. Since the pneumatic tyre compresses radially under the weight of the car, its radius is actually smaller than the unloaded radius at the point of contact, where it sticks and acquires linear velocity equal in magnitude and opposite in direction to the hub velocity. Since smaller wheels spin faster than larger ones at the same speed, the pneumatic tyre spins faster than the equivalent rigid tyre of the same unloaded radius. Let the unloaded, natural radius of the pneumatic tyre be R, also the radius of the equivalent solid tyre. If the hub has velocity V, the solid tyre spins with angular velocity $\omega = V/R$. Since the ***loaded radius***, of the pneumatic tyre, R_l, is smaller than R, V/R_l, the angular velocity of the loaded pneumatic tyre, must be larger than V/R."

This is partly correct. The pneumatic tyre-wheel combination *does* spin faster than a rigid wheel of the same unloaded radius, but it does *not* spin as fast as a rigid wheel of the same *loaded* radius, which is the height of the hub centre off the ground under load. The reason is that the tyre also compresses *circumferentially* or *tangentially*, setting up complex longitudinal twisting in the sidewall. The tangential speed of a particle of tread varies as the particle goes around the circumference of the tyre.

Let's mentally follow a piece of tread around as the *wheel*, not necessarily the tyre, turns at a constant radial velocity, ω_0. Imagine a plug of yellow rubber embedded in the tread, so that you could visually track it or photograph it with a movie camera or strobe system as it moves around the circumference. The rubber of the tread does not travel at constant speed, even though the wheel supporting the tyre does. At the top of the tyre, the radius is almost exactly *R*, the unloaded radius, so the tread moves with tangential velocity $R\omega_0$. As the yellow plug rolls around and approaches the contact patch from the front, it slows down in the bunched up area at the *leading edge* of the contact patch—just forward of it. There *is* a bunched-up area, because the tyre is made up of elastic material that gets squeezed and stretched out of the contact patch and piles up ahead of the contact patch as it rolls into it from the direction of the leading edge. Eventually, the plug enters the patch, in the centre of which it must move at speed $R_l\omega_0$ relative to the hub centre, that is, backwards at a speed dictated by the *loaded* radius and the wheel velocity. We've assumed that the plug is not slipping on the ground at the point where it has speed $R_l\omega_0$ with respect to the hub. This means that it has speed zero with respect to the ground at that point.

The average of the tangential velocities around the wheel defines the effective radius, R_e, as follows. Let θ measure the angular position, from 0 to 2π, around the wheel. Suppose we knew the tangential velocity with respect to the hub centre, $V(\theta)$, at every θ. We could easily measure this with our strobe light and cameras. $V(\theta)$ gives us the radius at every angular position via the equation $V(\theta)/\omega_0 = R(\theta)$, where ω_0 is the constant angular velocity of the wheel. The average would be computed by the following integral:

$$R_e = \frac{1}{2\pi}\int_0^{2\pi} R(\theta)\,d\theta = \frac{1}{2\pi\omega_0}\int_0^{2\pi} V(\theta)\,d\theta$$

Let's run some numbers. 10 mph is $14\frac{2}{3}$ feet/second or 176 inches/second. With an *unloaded* circumference of 26π inch/revolution, we get $176/26\pi = 2.154$ revs per second, or 129 RPM for each 10 MPH. Under ordinary circumstances, the effective radius will be no more than a few percent less than then the unloaded radius, and the RPMs should be, then, a few percent more than 129 RPM per 10 MPH. At 100 MPH, the tyre is under considerable stress and spins at something over 1,300 RPM.

Now we're in a position to define longitudinal slip, written σ. We want a quantity that vanishes when the wheel rolls at constant speed, increases when the wheel accelerates the car by pulling the contact patch backwards, and decreases below zero when then wheel brakes the car by pushing the contact patch forward. Under acceleration, the wheel and tyre combination will tend to spin a little faster than it would do while free rolling. We already know that, for a given *V*, the free-rolling angular velocity is $\omega_0 = V/R_e$, by definition. The *actual* angular velocity, ω, then, is higher under acceleration. So, if we know *V*, ω, and the constant R_e, then we can define the longitudinal slip as the ratio, minus 1, so that it's zero under free-rolling conditions:

$$\sigma = \frac{\omega}{\omega_0} - 1 = \frac{\omega}{V/R_e} - 1 = \frac{\omega R_e - V}{V}$$

Just looking at this formula, a free-rolling wheel has $\omega = \omega_0$, $\sigma = 0$, a locked-up wheel under braking has $\omega = 0$, $\sigma = -1$, and an accelerating wheel has a positive σ of any value.

The magic formula yields the longitudinal force, in Newtons, given some constants and dynamic inputs. The formula takes eleven empirical numbers that characterize a particular tyre $\{b_0, b_1, \cdots, b_{10}\}$. The dynamic parameters are , F_z, or *weight*, in **KiloNewtons** on the tyre, and the instantaneous slip, σ. The eleven numbers are measured for each tyre. We borrow an example from *Motor Vehicle Dynamics* by Giancarlo Genta. On page 528, he offers the following numbers for a car that appears to be a Ferrari 308 or 328, to which I have added dimensions:

b_0	1.65	dimensionless
b_1	0	1/MegaNewton
b_2	1688	1/Kilo
b_3	0	1/MegaNewton
b_4	229	1/Kilo
b_5	0	1/KiloNewton

b_6	0	$1/(\text{KiloNewton})^2$
b_7	0	1/KiloNewton
b_8	-10	dimensionless
b_9	0	1/KiloNewton
b_{10}	0	dimensionless

Though the majority of these values are zero for the tyres on this car, it is by no means always the case. In fact, the 'large-saloon' example just before the (alleged) Ferrari in Genta's book has *no* zeros.

We build up the magic formula in stages. The first helper quantity is $\mu_p = b_1 F_z + b_2$. This is an estimate of the ***peak, longitudinal coefficient of friction***, fitted as a linear function of weight (see Part 7 of PhORs). From this definition, we begin to see what's going with the dimensions. A typical, streetable sports car might weigh in at 3,000 lbs, which is about 3,000/2.2 = 1,500 * 0.9 = 1,350 kg, which is about 1,350 * 9.8 = 13,200 Newtons, or 13.2 KiloNewtons (look, ma, no calculator!). Let's assume each tyre gets a quarter of that to start off with, or 3.3 KN. b_1 multiplies that number to give us something with dimensions of KiloNewton/MegaNewton, which we write simply as 1/Kilo (inventing units on-the-fly, one Mega = 1 Kilo squared). b_2 has the same dimensions, so it's kosher to add it in, yielding $\mu_p = 1688/\text{Kilo}$ in this case. The next step is the helper $D = \mu_p F_z$, which will be in Newtons. We now see the reason for the 1/Kilo unit. In our case, we get about *D* = (1700-12)*3.3 = 5610-40 = 5570 N. The important point is that *D* **is linear in** F_z, so μ_p acts, mathematically, like a coefficient of friction, as promised. b_2 is a pretty direct measurement of stickiness, times 1,000 for convenience. This model tyre has a coefficient of friction of almost 1.7! Not my data, man.

The next step is to compute the product of a new helper, *B*, times b_0 and the aforecomputed *D*. The magicians who created the formula tell us that $Bb_0D = \left(b_3 F_z^2 + b_4 F_z\right)\exp\left(-b_5 F_z\right)$. This slurps up a few more of the magical eleven empirical numbers, and a pattern emerges. These b_i numbers serve as coefficients in polynomial expressions over F_z. So, b_5F_z is dimensionless, as must be the argument of the exponential function. $b_3F_z^2 + b_4F_z$ has dimensions of Newtons, as does the entire product. Therefore, *B* must be dimensionless. We need *B* in the next step, so let's solve for it now:

$$B = \frac{[Bb_0D]}{b_0D} = \frac{[Bb_0D]}{b_0\mu_pF_z} = \frac{(b_3F_z + b_4)\exp(-b_5F_z)}{\left[b_0\mu_p = b_0(b_1F_z + b_2)\right]},$$

Where we've been able symbolically to divide out one factor of F_z, convenient especially for numerical computation, where overflow is an ever-present hazard. Continuing with our numerical sample, $b_3F_z + b_4 = 229/\text{Kilo}$, the exponential is unity, and the numerator is

$$(b_0 = 1.65) * \left(\mu_p = \frac{1688}{\text{Kilo}}\right) \cong \frac{1688 + 844 + 169 + 85}{\text{Kilo}} = \frac{2786}{\text{Kilo}}$$

yielding $B = 229/2786 = 0.0822$. Most importantly, *B* **depends only weakly on** F_z. In the sample case, not at all, because $b_1 = b_3 = b_5 = 0$, but there are lots of other ways to characterize the algebraic dependence of *B* on F_z.

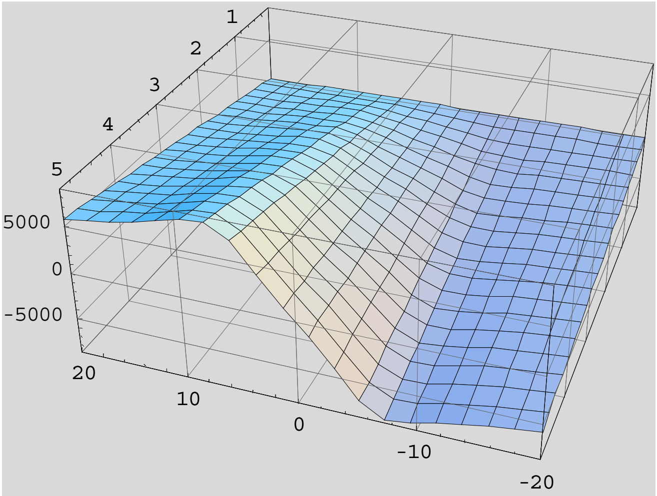
The next step is to account for the longitudinal slip with another helper, $S = (100\ \sigma + b_9F_z + b_{10})$; in our sample case, this reduces to just $S = 100\ \sigma$.

Only one more helper is needed, and that's $E = \left(b_6F_z^2 + b_7F_z + b_8\right)$, very straightforward. The final formula is

$$F_x = D\sin\left(b_0\ \tan^{-1}\left\{SB + E\left[\tan^{-1}(SB) - SB\right]\right\}\right)$$

Once again, don't try to find any physics in here: it's just a convenient formula that fits the data reasonably well. Plugging in numbers for $\sigma = 0$, because that's an easy sanity check to do in our heads, we see immediately the result is zero. Let's try $S = 10$, ten percent slip. $SB = 0.822$, $\tan^{-1}(0.822) = 0.688$, $E = -10$, so the argument of the outer arctangent is $SB - 10 * (-0.266) = SB + 2.66 = 3.48$, $\tan^{-1}(3.48) = 1.29$, $1.29\,b_0 = 2.13$, $\sin(2.13) = 0.848$, and, finally, $D * 0.848 = 4720$ Newtons. Lots of longitudinal force for a 3,300 N vertical load!

Let's plot the whole formula:



The horizontal axis measures $S = 100\,\sigma$, which is really just slip in percent. The deep axis, going into the page, measures F_z from 5 KN, nearest us, to zero in the back. The vertical axis measures the result of applying the formula to our model tyre, so it's longitudinal force—force of launching or braking. Notice that for a load of 5 KN, the model tyre can generate almost 8 KN of force. Very sticky tyre, as we've already noticed! Also notice that the generated force peaks at around $\sigma = 0.08$, or 8 percent. The peak would be something one could definitely feel in the driver's seat. Overcooking the throttle or brakes would produce a palpable reduction in g-forces as the tyres start letting go. Worse than that, increasing braking or throttle beyond the peak leads to reduced grip. This is an instability area, where *increasing slip leads to decreasing grip*.

Finally, note that the function behaves roughly linearly with F_z, showing that it acts like a Newtonian coefficient of friction, albeit a different one for each value of slip.

Physics of Racing, Part 22: The Magic Formula: Lateral Version

Brian Beckman, PhD

In this instalment, we review the other side of the magic formula: the one that computes lateral or cornering forces from slip angles (or grip angles). This formula is sufficiently similar to the longitudinal version of Part 21 that we can skip many preliminaries. But it's sufficiently different as to require careful exposition, leading us to define coordinate frames that will serve us throughout the rest of the *Physics of Racing* series. This instalment will be one to keep on hand for future reference.

Diving right in, just like its longitudinal sibling, this formula requires some magical constants, fifteen of them this time. Again, from Genta's possible-Ferrari data sheet:

a_0	1.799	dimensionless
a_1	0	1/MN
a_2	1688	1/Kilo
a_3	4140	N
a_4	6.026	KN
a_5	0	1/Degree
a_6	−0.3589	KN

a_7	1	dimensionless
a_8	0	dimensionless
a_9	−6.111/1000	Degree/KN
a_{10}	−3.224/100	Degree
$a_{11,1}$	0	1/MN-Degree
$a_{11,2}$	0	1/KiloDegree
a_{12}	0	1/Kilo
a_{13}	0	N

where N is Newton, KN is KiloNewton, and MN is MegaNewton. As with the longitudinal magic formula, there are lots of zeros in this particular sample case, but let us not confuse particulars with generalities. The formula can account for much more general cases.

The first helper is the peak, lateral friction coefficient $\mu_{yp} = a_1 F_z + a_2$, measured in inverse Kilos if F_z is in KN. Next is $D = \mu_{yp} F_z$, which is a factor *with the form of* the Newtonian model: normal force times coefficient of friction. In our sample, a_1 is zero, so μ_{yp} acts exactly like a Newtonian friction coefficient. In all cases, we should expect $a_1 F_z$ to be much smaller than a_2 so that it will be, at most, a small correction to the Newtonian behaviour.

To get the final force, we correct *D* with the following empirical factor:

$$\sin(\tau),\ \tau = a_0 \tan^{-1}(\upsilon)$$
$$\upsilon = SB + E\left[\tan^{-1}(SB) - SB\right]$$

This has exactly the same form as the empirical correction factor in the longitudinal version, but the component pieces, *S*, *B*, and *E* are different, here.

$$S = \alpha_{\text{degrees}} + a_8 \gamma_{\text{degrees}} + a_9 F_z + a_{10}$$

where α is the slip angle and γ is the camber angle of the wheel. In practice, we must carefully account for the algebraic signs of the camber angles so that the forces make sense at all four wheels. The usual negative camber, by the 'shop' definition, as measured on the wheel-alignment machine, will generate forces in the positive Y-direction on the right-hand side of the car and in the negative Y-direction on the left-hand side of the car. This comment makes much more sense after we've covered coordinate frames, below.

As before, we get *B* from a product, albeit one of greatly different form

$$Ba_0 D = a_3 \sin\left[2\tan^{-1}(F_z/a_4)\right]\left(1 - a_5|\gamma|\right)$$

where $|\gamma|$ is the absolute value of the camber angle, that is, a positive number no matter what the sign of γ. This gives

$$B \ = \ \frac{Ba_0 D}{a_0 D} \ = \ \frac{a_3 \sin\left[2\tan^{-1}(F_z/a_4)\right]\left(1 - a_5|\gamma|\right)}{a_0 \mu_{yp} F_z}$$

Almost done; include $E = a_6 F_z + a_7$ and sneak in an additive correction for ***ply steer*** and ***conicity***, which we'll leave undefined in this article:

$$S_v \ = \ \left[\left(a_{11,1} F_z + a_{11,2}\right)\gamma + a_{12}\right] F_z + a_{13}$$

To arrive at the final formula

$$F_y \ = \ D \sin\left(a_0 \tan^{-1}\left\{SB + E\left[\tan^{-1}(SB) - SB\right]\right\}\right) + S_v$$

This form is almost identical—in form—to the longitudinal version of the magic formula. The individual subcomponents are different in detail, however.

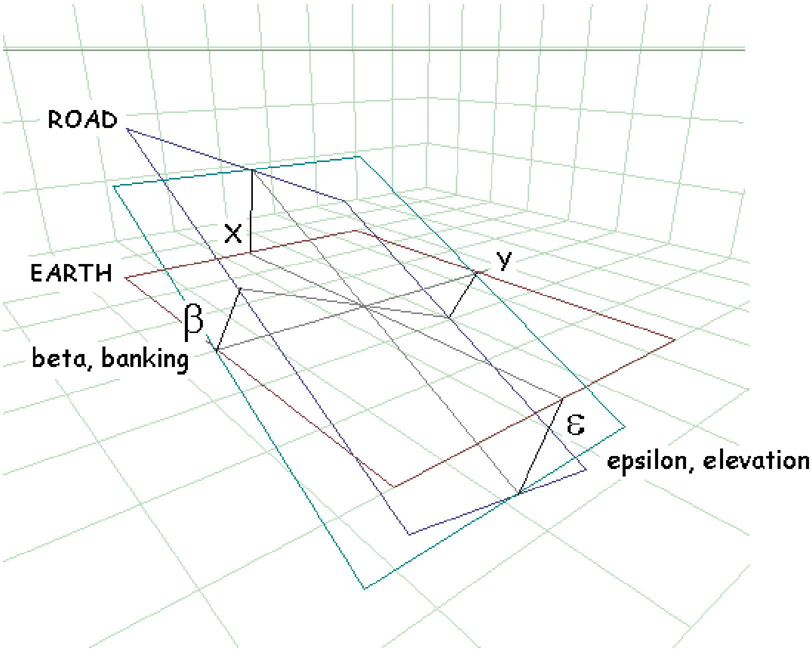
The most important input is the slip angle, α. This is the difference between the actual pathline of the car and the angle of the wheel. To be precise, we must define coordinate systems. We'll stay close to the conventions of the Society of Automotive Engineers (SAE), as published by the Millikens in *Race Car Vehicle Dynamics*. Note that this may differ from some frames we've used in the past, but we're going to stick with this set. There's a lot of intense verbiage in the following, but it's necessary to define precisely what we mean by wheel orientation in all generality. Only then can we measure slip angle as the difference between the path heading of the car and the wheel orientation.

First, is the ***EARTH*** frame, whose axes we write as $\{X,Y,Z\}$. The Z axis is aligned with Earth's gravitation and points *downward*. The origin of EARTH is fixed w.r.t. the Earth and the X and Y axes point in arbitrary, but fixed, directions. A convenient choice at a typical track might be the centre of start/finish with X pointing along the direction of travel of the cars up the main straight. All other coordinate frames ultimately relate back to EARTH, meaning that the location and orientation of every other frame must be given w.r.t. EARTH, directly or indirectly.

The next coordinate frame is ***CAR***, whose axes we write as $\{x, y, z\}$. This frame is fixed w.r.t. the sprung mass of the car, that is the body, with x running from tail to nose, y to driver's right, and z downward, roof through seat. Its instantaneous orientation w.r.t. EARTH is the ***heading,*** ψ. Precisely, consider the line formed by the intersection of EARTH's XY plane with CAR's xz plane. The angle of the line w.r.t. EARTH's X axis is the instantaneous heading of the car. It becomes undefined only when the car it points directly up—standing on its tail—or directly down—standing on its nose. To emphasize, ***heading is measured in the EARTH frame***.

The next coordinate frame is ***PATH***. The velocity vector of the car traces out a curve in 3-dimensional space such that it is tangent to the curve at every instance. The X-direction of PATH points along the velocity vector. The Z-direction of PATH is at right angles to the X direction and in the plane formed by the velocity vector and the Z-direction of EARTH. The Y direction of PATH completes the frame such that XYZ form an orthogonal, right-handed triad. The path of the car lies instantaneously in the XY plane of PATH. PATH ceases to exist when the car stops moving. ***Path heading*** is the angle of the projection of the velocity vector on EARTH's XY w.r.t. the X-axis of EARTH. Milliken calls this *course angle*, υ (Greek upsilon). Path heading, just like heading, is measured in the EARTH frame. The ***sideslip angle of the entire vehicle*** is the path heading minus the car heading, $\upsilon-\psi$. This is positive when the right side of the car slips in the direction of travel.

The next set of coordinate frames is $\boldsymbol{ROAD}_i$, where i varies from 1 to 4; there are four frames representing the road under each wheel, numbered as 1=Left Front, 2=Right Front, 3=Left Rear, 4=Right Rear. Each ROAD_i is located at the force centre of its corresponding contact patch at the point $\mathbf{R}_i \equiv \left(R_i^X, R_i^Y, R_i^Z\right)$ w.r.t. EARTH. This point moves with the vehicle, so, more pedantically, the origin of ROAD_i is $\mathbf{R}_i(t)$ written as a function of time. To get the X and Y axes of ROAD_i, we begin with a temporary, flat, coordinate system called $\boldsymbol{TA}_i$ aligned with EARTH and cantered at $\mathbf{R}_i$, then elevate by an angle $-90° < \varepsilon < 90°$, to get temporary frame $\boldsymbol{TB}_i$, and bank by an angle $-90° < \beta < 90°$, in that order, as illustrated below:



Consider any point P in space with coordinates $\mathbf{P} \equiv \left(P^X, P^Y, P^Z\right)$ w.r.t. EARTH. A little reflection reveals that its location w.r.t. TA_i is $\mathbf{P_{A\mathit{i}}} \equiv \mathbf{P} - \mathbf{R}_i$, just subtracting coordinates component-by-component. To get coordinates in TB_i, we multiply by the ***orthogonal matrix*** (once again, see www.britannica.com for brush-up) that does not change the Y components, but increases the Z and decreases the X components of points in the first quadrant for small, positive angles, namely:

$$\begin{pmatrix} \cos\varepsilon & 0 & -\sin\varepsilon \\ 0 & 1 & 0 \\ \sin\varepsilon & 0 & \cos\varepsilon \end{pmatrix}$$

We pick this matrix by inspection of the figure above or by application of the right-hand-rule (yup, see Britannica) Finally, to bank the system, we need the orthogonal matrix that does not change the X components, but increases the Y and decreases the Z components of first-quadrant points for small, positive angles, namely:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{pmatrix}$$

In case you missed it, we snuck in a reliable, seat-of-the-pants method for getting the signs of orthogonal matrices right. In any event, given $\mathbf{P}$ and $\mathbf{R}_i$, we compute the coordinates, $\mathbf{P_{R\mathit{i}}}$, of the point P in ROAD_i as follows:

$$\begin{pmatrix} \mathbf{P}^X_{\mathbf{R}i} \\ \mathbf{P}^Y_{\mathbf{R}i} \\ \mathbf{P}^Z_{\mathbf{R}i} \end{pmatrix} = \begin{pmatrix} \cos\varepsilon & 0 & -\sin\varepsilon \\ \sin\beta\sin\varepsilon & \cos\beta & \sin\beta\cos\varepsilon \\ \cos\beta\sin\varepsilon & -\sin\beta & \cos\beta\cos\varepsilon \end{pmatrix} \begin{pmatrix} \mathbf{P}^X - \mathbf{R}^X_i \\ \mathbf{P}^Y - \mathbf{R}^Y_i \\ \mathbf{P}^Z - \mathbf{R}^Z_i \end{pmatrix}$$

If the angles are small, $\cos\xi \approx 1$, $\sin\xi \approx \xi$, and the matrix can be simplified to

$$\begin{pmatrix} \mathbf{P}^X_{\mathbf{R}i} \\ \mathbf{P}^Y_{\mathbf{R}i} \\ \mathbf{P}^Z_{\mathbf{R}i} \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\varepsilon \\ \beta\varepsilon & 1 & \beta \\ \varepsilon & -\beta & 1 \end{pmatrix} \begin{pmatrix} \mathbf{P}^X - \mathbf{R}^X_i \\ \mathbf{P}^Y - \mathbf{R}^Y_i \\ \mathbf{P}^Z - \mathbf{R}^Z_i \end{pmatrix}$$

Even at 20 degrees, the errors are only about 6% in the cosine and 2% in the sin, resulting in a maximum error of 12% in the lower right of the matrix. This matrix approximation is suitable for the majority of applications. One feature of orthogonal matrices is that their *inverse* is their *transpose*, that is, the matrix derived by flipping everything about the main diagonal running from upper left to lower right. In the small-angle approximation, we get

$$\begin{pmatrix} 1 & 0 & -\varepsilon \\ \beta\varepsilon & 1 & \beta \\ \varepsilon & -\beta & 1 \end{pmatrix} \begin{pmatrix} 1 & \beta e & \varepsilon \\ 0 & 1 & -\beta \\ -\varepsilon & \beta & 1 \end{pmatrix} = \begin{pmatrix} 1+\varepsilon^2 & 0 & 0 \\ 0 & 1+\beta^2\left(1+\varepsilon^2\right) & \beta\varepsilon^2 \\ 0 & \beta\varepsilon^2 & 1+\beta^2+\varepsilon^2 \end{pmatrix}$$

The right-hand side is very close to the unit matrix because the squares of small angles are smaller, yet. With the inverse matrix we can convert from coordinates in ROAD_i to coordinates in EARTH:

$$\begin{pmatrix} \mathbf{P}^X \\ \mathbf{P}^Y \\ \mathbf{P}^Z \end{pmatrix} = \begin{pmatrix} 1 & \beta\varepsilon & \varepsilon \\ 0 & 1 & -\beta \\ -\varepsilon & \beta & 1 \end{pmatrix} \begin{pmatrix} \mathbf{P}^X_{\mathbf{R}i} \\ \mathbf{P}^Y_{\mathbf{R}i} \\ \mathbf{P}^Z_{\mathbf{R}i} \end{pmatrix} + \begin{pmatrix} \mathbf{R}^X_i \\ \mathbf{R}^Y_i \\ \mathbf{R}^Z_i \end{pmatrix}$$

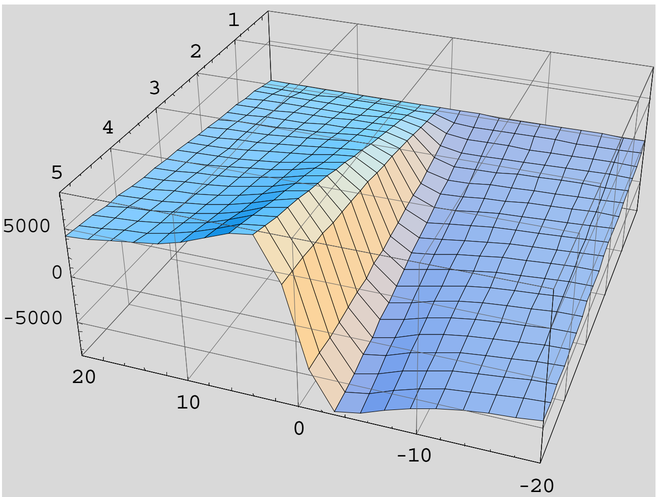
The last set of coordinate frames is ***WHEEL$_i$***. As with ROAD_i, there is one instance per wheel. WHEEL_i is cantered at the wheel hub. Under normal rolling, the coordinates of its origin in ROAD_i are $\mathbf{W}_{\mathbf{R}i} \equiv \left(0,0,-R_l\right)$, where R_l is the loaded radius of the tyre-wheel combination. Pedantically, R_i should be corrected for elevation and banking, but such corrections would be small for ordinary angles—on the order of $2-\cos\beta\cos\varepsilon$ —plus it seems *not* to be standard practice (I can find no reference to it in my sources). More important is the orientation of WHEEL_i. Consider the plane occupied by the wheel itself. This plane intersects ROAD_i in a line that defines the X direction of WHEEL_i, with the positive direction being as close to that of travel as possible. The Y direction points to driver's right. The wheel plane is tilted by a camber angle, γ, about the X-axis of the WHEEL coordinate system. To emphasize: WHEEL_i ***does not include wheel camber***, and it differs from

ROAD only by a rotation about ROAD's Z axis that accounts for the pointing direction of the wheel.

At this point, you should create a mental picture of these coordinate frames under typical racing conditions. Picture a CAR frame yawed at some heading w.r.t. EARTH—and perhaps pitched and rolled a bit; a PATH frame aligned at some slightly different path heading; and individual ROAD and WHEEL frames under each tyre contact patch, where the ROAD frames are perhaps tilted a bit w.r.t. EARTH and the WHEEL frames are aligned with the wheel planes but coplanar with the ROAD frames. For a car travelling on a flat road at a stable, flat attitude, the XY planes of CAR, PATH, and EARTH would all coincide and would differ from one another only in the yaw angles ψ and υ. When some tilting is engaged, ψ and υ are still defined by the precise projection mechanisms explained above.

Now, imagine the X-axis of CAR projected on the XY plane of each WHEEL frame and translated—without changing its direction—to the origin of WHEEL. The angle of WHEEL's X axis, which is the same as the plane containing the wheel, w.r.t. the projection of CAR's X axis, defines the steering angle, δ, of that wheel. Finally, imagine PATH's X axis projected onto the XY plane of WHEEL in exactly the same way. Its angle w.r.t. to the X axis of WHEEL, in all generality, defines the slip angle. Since WHEEL is lilted w.r.t. gravitational *down*, the load, F_z, on the contact patch, which we need for the magic formula, must be computed in WHEEL. It will be smaller than the total weight, W_i, by factors of $\cos\beta$ and $\cos\varepsilon$, which are obviously unity under the small-angle approximation.

At last, we can plot the magic formula:



The horizontal axis measures slip angle, in degrees. The vertical axis measures lateral, cornering force, in Newtons. The deep axis measures vertical load on the contact patch, in KiloNewtons. We can see that these tyres have a peak at about 4 degrees of slip and that cornering force goes *down* as slip goes up on either side of the peak. On the high side of the peak, we have dynamic understeer, where turning the wheel more makes the situation worse. This is a form of instability in the control system of car and driver.

As a final comment, let me say that I am somewhat dismayed that the magic formula does *not* account for any variation of the lateral force with speed. Intuitively, the forces generated at high speeds must be greater than the forces at low speed with the same slip angles. However, the literature—sometimes explicitly, and sometimes by sin of omission—states that the magic formula doesn't deal with it. One of the reasons is that, experimentally, effects of speed are extremely difficult to separate from effects of temperature. A fast-moving tyre becomes a hot tyre very quickly on a test rig. Another reason is that theoretical data is usually closely guarded and is not likely to make it into a consensus approximation like the magic formula. This is a fact of life that we hope will not affect our analyses too adversely from this point on.

The Physics of Racing,
Part 23: Trail Braking

Brian Beckman, PhD

Trail-braking is a subtle driving technique that allows for later braking and increased corner entry speed. The classical technique is to complete braking before turn-in. This is a safer, easier technique for the driver because it separates traction management into two phases, braking and cornering, so the driver doesn't have to chew gum and walk at the same time, as it were. With the trail-braking technique, the driver carries braking into the corner, gradually trailing off the brakes while winding in the steering. Since braking continues in the corner, it's possible to delay its onset in the preceding straight braking zone. Since it eliminates the sub-optimal moments between the ramp-down from braking and the ramp-up to limit cornering by *overlapping* them, entry speeds can be higher. The combination of these two effects means that the advantage of later braking is carried through the first part of the corner. In many ways, this is the flip side to corner exit, where any speed advantage due to superior technique gets carried all the way down the ensuing straight. The magnitude of the trail-braking effect is much smaller, though: perhaps a car length or two for a typical corner. Done consistently, though, it can accumulate to whole seconds over a course.

When I was taught to drive in the '80s, not all the fast drivers used trail braking and instructors usually gave it at most a passing mention as an optional, advanced technique. The reason was probably a risk-benefit analysis:

- it's a small effect compared to the big-picture basics, like carrying speed *out* of a corner, that everyone must learn early on
- it's difficult to learn, so why burden new students with it?
- mistakes with it are ugly

Another reason may have been that my instructors hadn't got their butts kicked recently by a trail-braking driver. It was not a commonplace technique back then, so one might drive a whole season of club racing without getting spanked by trail braking. Since not everyone used it, not everyone had to develop the skill.

Nowadays, however, the general level of driving skill has increased to the point where it's no longer optional, unless you're content with fourth place.

As with most driving skills, it's difficult to get a feel for the limits without exceeding them from time to time. However, exceeding the limits at trail braking has some of the worst consequences one can invite on a race track, typically worse than those from mistakes at corner exit. It's definitely a big risk for a small effect, justified only because it accumulates. Blowing it results in *too high an entry speed*. You get:

- inappropriate angular attitude in the corner
- immediate probing of the understeer or oversteer characteristics of the car

- surprise, pop quiz on the driver's car-control skills
- missed apex and track-out points
- a looming penalty cone, gravel trap, tyre barrier, concrete wall, tree, etc.
- when you bounce back from *that* impact, you can hit other cars, spectators, corner marshals, berms, etc.
- anything else that can go wrong in a blown corner

That's one of the reasons I have not, in the past, singled it out for my personal driver-development work - it's hard to do at all and harder to do it consistently and just didn't seem worth it. Another reason is that the kinds of cars I like to drive let you get away without it much of the time. I prefer ultra-powerful cars because they're fun and loud and attract a lot of attention. Paradoxically, though, such cars can lull one into becoming a lazy driver. With a lot of power on tap, you can often make up for an overly conservative entry speed on the exit.

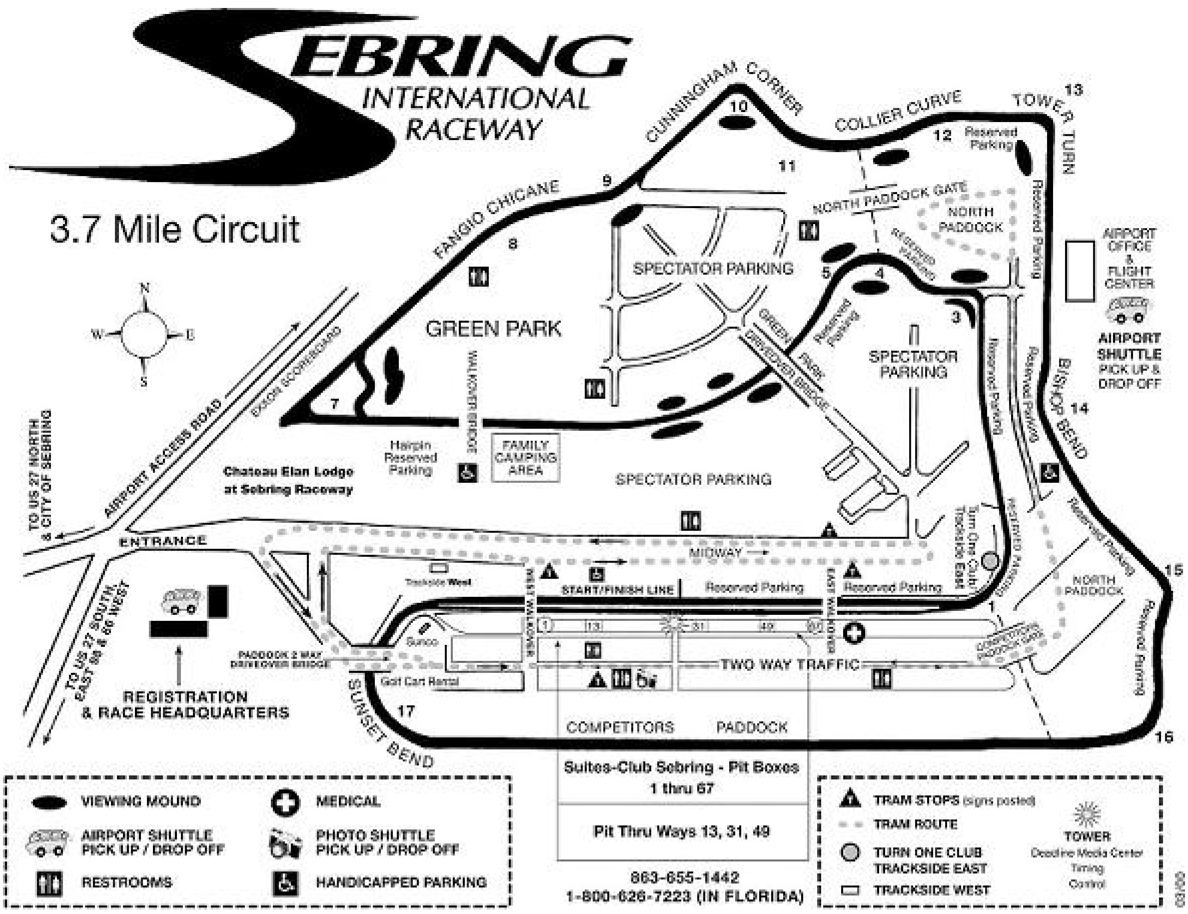
However, when the cars are equalized, as in spec races, showroom stock, or in a lot of Solo II car classes, trail braking takes a prominent role. It can be difficult to spot it as an issue in Solo II, where drivers are alone against the clock. All else being equal, a Solo II driver without trail braking may just find himself scratching his head wondering how in blazes the other drivers can be so much faster. Go wheel-to-wheel on the track with equal cars, though, and the issue becomes instantly and ***visually obvious***. You may be just as fast *in* the corner, coming *out* of the corner, *down* the straight. You may have perfect threshold braking. You may have perfect turn-in, apex and track out points. But that little extra later braking and entry speed will allow the trail-braker to take away several feet every corner. Corner after corner, lap after lap, he will gobble you up.

I recently completed a road-racing school at Sebring International Raceway where this is precisely what I saw. In identical Panoz school cars, the drivers who were faster than I were doing it right there and nowhere else. My ingrained, outdated style did me in, and even though I had much, much more on-track experience than the rest of the students, and even though they weren't faster in top speed than I, and even though their cornering technique was not nearly as polished as mine, three (out of twelve) of them had better lap times than I.

The instructors were as surprised as I. One even said he would have bet money that I was the quickest from watching me and riding with me (instructors did not ride in the wheel-to-wheel sessions). The clock doesn't lie though, and we were scratching our heads and I started swapping cars. Once we went wheel-to-wheel on the third day of the program, I spotted it, right there the first time into turn 2: the three quicker drivers took a car length from me on the corner entry. They did it again in turn 10 (Cunningham), at the Tower turn, and turn 15 approaching the back stretch: all the turns requiring full braking and downshifts. I made up a bit at the hairpin, which is an autocrosser's corner if there ever was one, and I knew the importance of not missing the apex by more than an inch or two if possible. They also couldn't beat me entering turn 17, which has no straight braking zone: instead, the best technique is to brake partially after turn in (at 115 mph, this is big-time, serious fun). Thus, turn 17 did not trigger my old-fashioned "braking-zone" program, and I was able to use my high-speed experience to coax a bit more than average grip through it. So, in sum, my conservative turn-ins on the slow corners added up to about half a second per lap, which is about 65 feet at the start-finish line where we're going about 90 mph =132 fps (90 x 22 / 15). Ugly.

I was doing it the old-fashioned way: get the braking done in the braking zone and get your foot back on the gas pedal and up to neutral throttle before turn-in. That little tenth of a second or so where I'm coasting and they're still braking *is* the car-length they were taking out on me. It was small enough that the instructors couldn't feel it or see it. But electronic instrumentation would have picked it up. When I go back to the Panoz Sebring school next year, I will take advanced sessions in fully instrumented cars, where the instructors go out for some laps at 10/10s, then the students go out in the same car and take data. Back in the pits, the charts are differenced and the student can see precisely what he needs to do to come up to the instructor's level (most of the instructors have years of experience on the track, and hold current or former lap records in various cars on the course, so it's quite unlikely that a student will be as quick out of the box).

The following is a picture of the course snipped from the web site at http://www.sebringraceway.com, so you can see the bits of the course I'm talking about:



Let me say a few things about the school. The three-day program consisted of

- solo exercises in braking, skid recovery, and autocrossing
- detailed in-car instruction as driver and passenger over several lapping sessions

- racecraft including passing and rolling starts
- wheel-to-wheel sessions on the full open course

It's a great program, easily better than spending the same amount of money on the car: highly recommended.

Sebring is large, exciting, lovely, complex course with a deep history of sports-car racing. It is currently 3.70 miles in length, though it has been as long as 5.7 miles in its history. Let's do some dead reckoning, that is, math in our heads without even envelopes to write on. We'll see if we can cook up some data, from memory, to justify the intuitions and explain the results above.

There are 2.54 centimetres per inch: that's an exact number. Therefore, there are 2.54 x 12 = 30.48 centimetres per foot. The number of centimetres per mile, then are 30.48 x 5280 = 30 x 52 x 100 + 30 x 80 + 48 x 52 + 48 x 80 / 100 = 156000 + 2400 + (50 - 2)(50 + 2) + 3840 / 100 = 158400 + 2500 - 4 + 38.40 = 160,934.4. Thus, a mile contains 1.609344 kilometres, which we can round to 1.61, which is, conveniently, 8/5 + 1/100. So 3.70 miles is 29.637 / 5 = 5.927 kilometres or just about 6. Now, there are 5280 / 3 = 1760 yards in a mile, so we have 3700 + 2590 + 222 = 6,512 yards, which is consistent with 6 kilometres, so we've got a check on our math. In fact, we can be a little more sanguine about it. Another number we remember is that there are about 39 inches per metre; that's a yard and three inches, or 13/12 yard. So, if we have about 6,000 metres, that's going to be about 6,000 + 6,000 / 12 = 6,500 yards. Amazing, isn't it? Finally, this is 6,512 x 3 = 13,036 + 6,512 = 19,048 feet.

Big Track. Nice.

A record time around the course in the Panoz school cars is 2 min 28 seconds. The students were doing 2:40 to 2:45. I believe I uncorked a 2:36 somewhere along the way, but my typical lap was 2:40 and the quicker guys pulled about 65 feet on me at the start-finish every lap, which I reckoned before to be worth half a second. What's the average speed at 2:40? That's 3.70 miles in 160 seconds. The average speed is 19,048 / 160 fps ~ 1905 / 16 ~ 476 / 4 ~ 119 fps, which is 119 x 15 / 22 mph, and that is (1190 + 595 ~ 1785) / 22 = 892.5 / 11. It's hard to divide by 11, so lets multiply instead. 80 mph by 11 would be 880, and that's not enough by 12.5. So, if we go with 81 mph by 11, namely 891, we're short by 1.5. A tenth of 11 will take care of some of that, so 81.1 by 11, namely 892.1, leaves us close enough. Now, doing the same distance in 2:28, or 148 seconds, yields an average speed of 19,048 / 148 ~ 4,762 / 37. Another tough divisor. Let's try 130 x 37 = 3700 + 1110 = 4810, too much by 48. But, we lucked out, it's obvious that 48 is about 1.30 x 37, so we get 130 - 1.30 = 128.7 fps. Now multiply that by 15 / 22: (1287 + 643.5) / 22 ~ 1930 / 22 = 965 / 11. 90 x 11 would be 990, too much by 25, which is a little more than 2 x 11. So 90 - 2 = 88 x 11 would be 880 + 88 = 968, too much by 3, so we'll reduce 88 by 0.3 x 11 to get 87.7. The average speed of a record-setting lap is 6.6 mph faster than our pitiful student laps! The difference is 12 seconds, so, as a rule of thumb, a second at 85 mph average is worth a little more than 1/2 an mph.

But, before we wander too far off topic, let's compare 2:40 to 2:40.5, since my contention from the beginning of this note is THAT difference can be accounted entirely to trail braking in four corners of this course: 2, 10, 13, and 15. Well, at 119 fps, average speed, half a second is about 60 feet, which is about 4 car lengths. Yep, there you have it: one car length per significant corner due to trail braking. Darn it, looks like I'll just have to go back there and keep trying, over and over again.

The Physics of Racing, Part 24: Combination Slip

Brian Beckman, PhD

The goal in this and the next instalment of the *Physics of Racing* is to combine the magic formulae of parts 21 and 22, so that we have a model of tyre forces when turning and braking or turning and accelerating *at the same time*. In this part, we figure out combination slip, and in the next instalment, we figure out combination grip. Roughly speaking, slip is the input and grip is the output to our model. Slip comes from control inputs on brakes, throttle and wheel, grip comes from reaction forces of the ground on the tyres.

The regular magic formulae apply only to a tyre generating longitudinal or lateral forces in isolation, that is, to a tyre accelerating or braking and *not* turning, or a tyre turning but *not* accelerating or braking. In part 7, we approximated the response under combination slip by noting that it follows the circle of traction. A tyre cannot deliver maximal longitudinal grip when it's delivering lateral grip at the same time, and *vice versa*. According to my sources, modelling of combination slip and grip is an area of active research, which means we are on our own, once again, in the original, risk-taking spirit of the *Physics of Racing* series. In other words, we're going way out on a limb and this could all be totally wrong, but I promise you lots of fun physics on the journey.

From part 21, recall our definition for the longitudinal slip, σ, the input to the longitudinal magic formula

$$\sigma = \frac{\omega}{\omega_0} - 1 = \frac{\omega}{V/R_e} - 1 = \frac{\omega R_e - V}{V}$$

where V is the forward speed of the hub w.r.t. ground, R_e is a constant, the ***effective radius***, for a given tyre, and ωR_e is the *backward* speed of the CP w.r.t. the hub. Therefore, $\omega R_e - V$ is the backward speed of the CP w.r.t. EARTH. A slick technique for proving this, and, in fact, for figuring out *any* combinations of relative velocities (see part 19) is as follows. Write $V = \mathrm{HUB} - \mathrm{GD}$, meaning speed of the HUB relative to the ground (GD). Now write $\omega R_e = -(\mathrm{CP} - \mathrm{HUB})$, meaning the backward speed of the CP relative to the HUB; the overall minus sign outside reminds us that we want ωR_e positive when the CP moves *backwards* w.r.t. the hub. Now, just do arithmetic:

$$\begin{aligned}\omega R_e - V &= -(\mathrm{CP} - \mathrm{HUB}) - (\mathrm{HUB} - \mathrm{GD}) \\ &= -\mathrm{CP} + \mathrm{HUB} - \mathrm{HUB} + \mathrm{GD} = -(\mathrm{CP} - \mathrm{GD})\end{aligned}$$

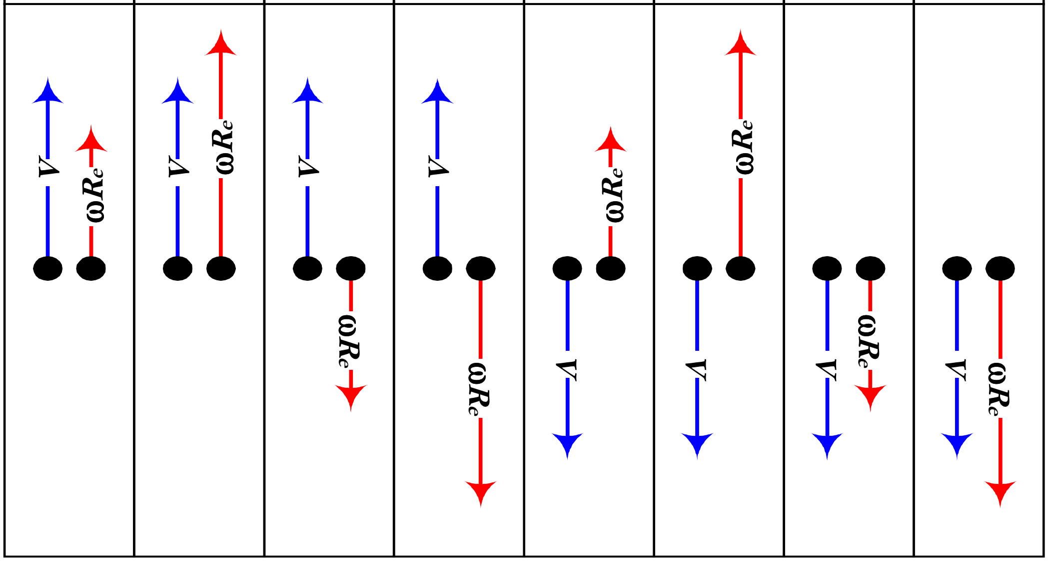
voila, backward speed of the CP w.r.t. the ground. This realization gives us intuition into the sign of σ: if and only if the CP moves backwards faster than the hub moves forward; the car accelerates forward—visualize that in your head; in that case, ωR_e is greater than V; $\omega R_e - V$ is greater than zero; and σ is positive.

It turns out that we developed this formula only for the case when V is positive, that is, the car is moving forward. And, in fact, the formula only works in that case. To generalize it to cars moving in reverse, we'd best analyse it in excruciating detail. A moment's reflection reveals that there are eight cases: two signs for V, two signs for ωR_e, and two cases for whether the absolute value of V is greater than the absolute value of ωR_e, yielding eight = $(2\times2\times2)$ possibilities, which have the following physical interpretations:

1. Car (hub) moving forward, CP moving slowly forward w.r.t. ground, resisting car motion. This models driving slowly in reverse gear while moving forward. ***Slowly***, here, means slowly relative to V, or, precisely, that $|\omega R_e| < |V|$, where $|\omega R_e|$ is the absolute value or unsigned magnitude of ωR_e and $|V|$ is the absolute value or unsigned magnitude of V.
2. Same as above, just with CP moving ***quickly*** forward, that is with $|\omega R_e| > |V|$.
3. Car moving forward, CP moving slowly backwards, just not quickly enough to accelerate the car. This is braking or engine braking in forward gear. Wheel lockup while moving forward falls in this case, too.
4. Car moving forward, CP moving quickly backwards, accelerating the car forward.
5. Car moving backward, CP moving slowly forwards, just not quickly enough to accelerate the car backward. This is braking or engine braking in reverse, and wheel lockup in reverse falls in this case.
6. Car moving backward, CP moving quickly forward, accelerating in reverse.
7. Car moving backward, CP moving slowly backward, resisting motion.
8. Car moving backward, CP moving quickly backward, resisting motion.

We've caught all this in the following diagram, in which we have drawn V and ωR_e as arrows, pointing in the actual direction that the hub moves w.r.t. the ground and the CP moves w.r.t. the hub, respectively. Algebraically, V and ωR_e have opposing sign conventions, so ωR_e is negative when its arrow points up. In looking at this table, note that the longitudinal force F_x has the same sign as σ. When F_x and σ are positive, the car is being forced forward by the ground's reacting to the tyres. When they're negative, the car is being forced backwards. So, to figure out which way the car is being forced, just look at the sign of σ.

case 1	case 2	case 3	case 4	case 5	case 6	case 7	case 8
Spinning Slowly Forward	Spinning Fast Forward	Spinning Slowly Backward	Spinning Fast Backward	Spinning Slowly Forward	Spinning Fast Forward	Spinning Slowly Backward	Spinning Fast Backward
Moving Forward				Moving Backward			
$V > 0$	$V > 0$	$V > 0$	$V > 0$	$V < 0$	$V < 0$	$V < 0$	$V < 0$
$\omega Re < 0$	$\omega Re < 0$	$\omega Re > 0$	$\omega Re > 0$	$\omega Re < 0$	$\omega Re < 0$	$\omega Re > 0$	$\omega Re > 0$
$\omega Re\text{-}V<0$	$\omega Re\text{-}V<0$	$\omega Re\text{-}V<0$	$\omega Re\text{-}V>0$	$\omega Re\text{-}V>0$	$\omega Re\text{-}V<0$	$\omega Re\text{-}V>0$	$\omega Re\text{-}V>0$
$\sigma < 0$	$\sigma < 0$	$\sigma < 0$	$\sigma > 0$	$\sigma > 0$	$\sigma < 0$	$\sigma > 0$	$\sigma > 0$



Inspection of this table reveals that the following *new* formula works in all cases:

$$\sigma = \frac{\omega R_e - V}{|V|}$$

Where the numerator, $\omega R_e - V$ is the *signed* difference of the two speeds and the denominator is *unsigned*. It is perhaps surprising that there is so much richness in such a little formula. However, it is precisely this richness that we must maintain as we add steering, that is, lateral slip angle at the same time. The best way to do that is to vectorise the formula so that the algebraic signs of the vector components take the place of the signed quantities V and ωR_e. The approach here parallels the approaches of parts 16 and 19. We want the signed component V_x to take the place of the old, signed V, the signed component L_x of the slip velocity $\mathbf{L}$ to take the place of the old ωR_e, and V *now* to denote the unsigned magnitude of the vector $\mathbf{V}$, that is $V \equiv \|\mathbf{V}\| \equiv +\sqrt{V_x^2 + V_y^2 + V_z^2}$. The next table summarizes these changes:

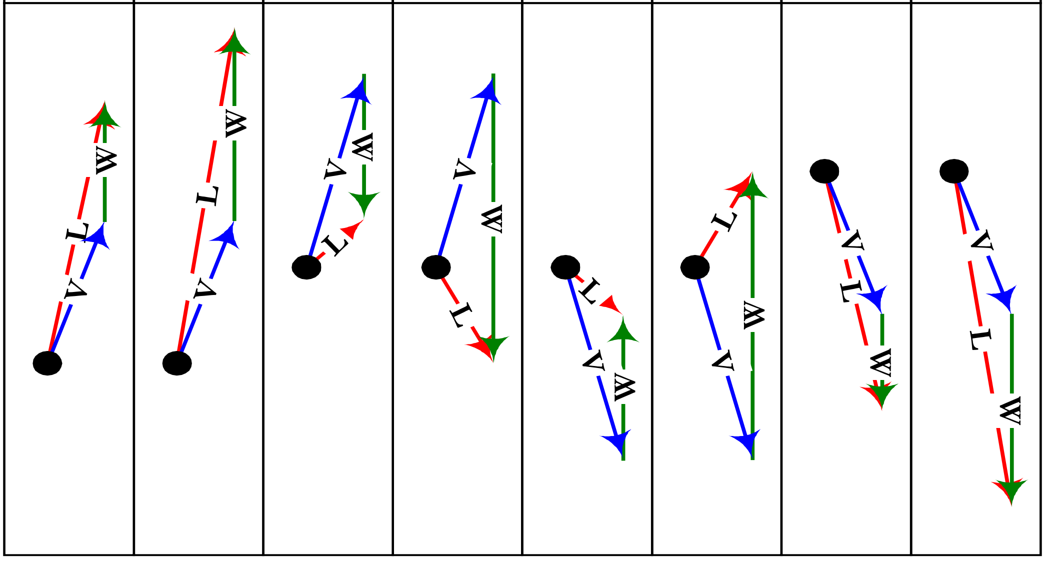
Quantity	*old notation*	*new notation*	*vector*
signed, forward speed of hub w.r.t. EARTH	V	V_x	$\mathbf{V}$
signed, backward speed of CP w.r.t. hub	ωR_e	$-W_x$	$\mathbf{W}$
unsigned magnitude of hub speed	$\lvert V \rvert$	V	$\lVert \mathbf{V} \rVert$ or V
signed, backward speed of CP w.r.t. EARTH	$\omega R_e - V$	$-L_x$	$\mathbf{L} = \mathbf{V} + \mathbf{W}$
signed, longitudinal slip	$\sigma = \dfrac{\omega R_e - V}{\lvert V \rvert}$	$\sigma = \dfrac{-L_x}{V}$	?

Slip velocity, $\mathbf{L} \equiv \left[L_x, L_y, L_z\right] = \mathbf{V} + \mathbf{W}$ is the plain-old vector velocity of the CP w.r.t. EARTH with no secret sign convention to confuse things. As an aside, we note that when the car sticks to the ground on flat road, we may assume $L_z = 0$. **W** is CP velocity w.r.t. hub. In the TYRE system, **W** has only a (signed) x-component, that is, $\mathbf{W}_{\text{TYRE}} = [W_x, 0, 0]$. These definitions hold whether the car is moving forward or backward, accelerating or braking.

The big question mark in the table indicates that we do not have a vector for combination slip because we measure its longitudinal and lateral components differently, as a ratio and as an angle, respectively. Note that, since lateral slip α is the angle made by **V** in the TYRE system, it is $\tan^{-1}\left(V_{y,[\text{TYRE}]} / V_{x,[\text{TYRE}]}\right)$. Since $\mathbf{L} = \mathbf{V} + \mathbf{W}$, it's easy to see that $\alpha = \tan^{-1}\left(\dfrac{L_y}{L_x - W_x}\right)$, which is a most convenient expression, though some attention must be paid to the quadrant in which the angle falls. We resolve this in the next two installments of *PhORS* as we stitch together the two magic formulae to make *Combination Grip*.

But first, let's update the big diagram, showing all eight cases with a little slip angle thrown into the mix, and the vector sum, **L** = **V** + **W**, replacing the *ad hoc*, signed quantities of the old notation. The sign of the slip angle α does not introduce new cases so long as $\lvert\alpha\rvert < 90^\circ$ because the right-hand and left-hand cases are precisely symmetrical. The nice thing, here, is that we can treat all eight cases the same way—the nature of vector math takes care of it because the magnitude of a vector is always unsigned. Using signed, scalar quantities, we had to dissect the system and introduce *absolute value* to get everything to work. Absolute value has always struck me as a kind of crock or kludge to use when the math is just not sufficiently expressive. The main contribution of this instalment is to fix that problem.

case 1	*case 2*	*case 3*	*case 4*	*case 5*	*case 6*	*case 7*	*case 8*
Spinning Slowly Forward	*Spinning Fast Forward*	*Spinning Slowly Backward*	*Spinning Fast Backward*	*Spinning Slowly Forward*	*Spinning Fast Forward*	*Spinning Slowly Backward*	*Spinning Fast Backward*
Moving Forward				*Moving Backward*			
$V_x > 0$	$V_x > 0$	$V_x > 0$	$V_x > 0$	$V_x < 0$	$V_x < 0$	$V_x < 0$	$V_x < 0$
$-W_x < 0$	$-W_x < 0$	$-W_x > 0$	$-W_x > 0$	$-W_x < 0$	$-W_x < 0$	$-W_x > 0$	$-W_x > 0$
$-L_x < 0$	$-L_x < 0$	$-L_x < 0$	$-L_x > 0$	$-L_x > 0$	$-L_x < 0$	$-L_x > 0$	$-L_x > 0$
$\sigma < 0$	$\sigma < 0$	$\sigma < 0$	$\sigma > 0$	$\sigma > 0$	$\sigma < 0$	$\sigma > 0$	$\sigma > 0$



The Physics of Racing, Part 25: Combination Grip

Brian Beckman, PhD

In this instalment of the *Physics of Racing*, we complete the program begun last time to combine the magic formulae of parts 21 and 22, so that we have a model of tyre forces when turning and braking or turning and accelerating at the same time. Parts 21 and 22 introduced the magic formulae. The first one takes longitudinal slip as input and produces longitudinal grip as output. The other one takes lateral slip as input and produces lateral grip. Slip depends primarily on driver inputs, grip is force generated at the ground. *Longitudinal* means in the straight-ahead direction. *Lateral* means sideways, as in the forces for turning. Since the magic formulae work only in isolation, we have work to do to model turning and braking at the same time and turning and accelerating at the same time.

Last time, we vectorised slip—the input—to come up with ***combination slip***, captured in the vector ***slip velocity***. That vector measures the velocity of the contact patch with respect to (w.r.t.) the ground in one, handy definition. This time, we first boil down combination slip to new inputs for the old magic formulae. In the old magic formulae, we measure longitudinal slip as a percentage of unity, that is, as a percentage of breakaway sliding; and we measure lateral slip as an angle in degrees. These are not ***commensurable***, meaning that we do not use the same units of measurement for both kinds of slip. That's why there was a big, fat question mark in the vector slot for combination slip in one of the tables in part 24. Once we make them commensurable, then we stitch the magic formulae together to get one vector gripping force as a function of one vector slip. This finally allows us to compute the forces delivered by a tyre under combination control inputs.

Once again, we are in uncharted territory, so take it all in the for-fun spirit of this whole series of articles. I don't represent anything I do here as authoritative racing practice. I only claim to be bringing the fresh perspective of a stubbornly naïve physicist to the problems of racing cars as an amateur. The standard practice of the professional racing engineering community may be completely different. This is the *Physics of Racing*, not the *Engineering of Racing*. I'm after the fundamental principles behind the game. I use techniques that may be foreign to the engineers that build and race cars professionally. My results may not be precise enough for final application. I may take approximations that simplify away things that are actually critically important. On purpose, I'm figuring things out on my own. Often, this helps me understand published engineering information better. Just as often, it helps me debunk and debug the conventional wisdom. If you find mistakes, gaffs, or laughable dumb stuff, or if you know better ways to do things, I encourage you to fire up debate, publish rebuttals, or write to me directly. I've done my best to track down the latest and greatest information, but I've found lots of errors, ambiguities, and inexplicabilities in the open literature. I also suspect a conspiracy, meaning that I'd bet that the tyre manufacturers and pro racing teams don't publish their best information—I certainly wouldn't if I were they.

Disclaimers out of the way, we now have enough tools on the table to combine the two magic formulae. Recall the formulae from parts 21 and 22: $F_x(\sigma, F_z)$ and $F_y(\alpha, F_z, \gamma)$ for the longitudinal and lateral forces. Here they are, in isolation:

$$\boxed{\begin{aligned}&F_x(\sigma, F_z) = D\sin\Big(b_0\tan^{-1}\Big\{SB + E\Big[\tan^{-1}(SB) - SB\Big]\Big\}\Big) + S_u\\ &\quad D = \mu_{xp}F_z = (b_1F_z + b_2)F_z,\quad B = (b_3F_z + b_4)\exp(-b_5F_z)\big/b_0(b_1F_z + b_2)\\ &\quad E = \big(b_6F_z^2 + b_7F_z + b_8\big),\qquad S = (100\,\sigma + b_9F_z + b_{10}),\quad S_u = \text{constant}\end{aligned}}$$

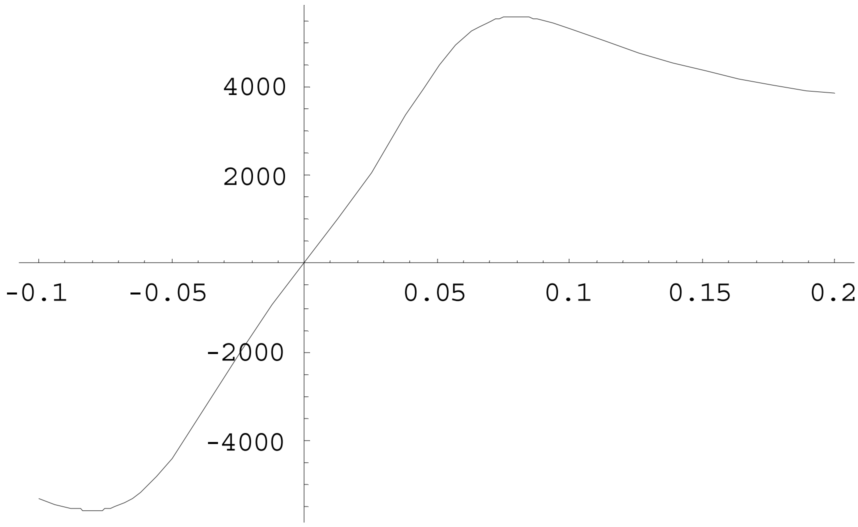
$$\boxed{\begin{aligned}&F_y(\alpha, F_z, \gamma) = D\sin\Big(a_0\tan^{-1}\Big\{SB + E\Big[\tan^{-1}(SB) - SB\Big]\Big\}\Big) + S_v\\ &\quad D = \mu_{yp}F_z = (a_1F_z + a_2)F_z,\quad B = a_3\sin\Big[2\tan^{-1}(F_z/a_4)\Big](1 - a_5|\gamma|)\big/a_0(a_1F_z + a_2)F_z\\ &\quad E = a_6F_z + a_7,\qquad S = \alpha_{\text{degrees}} + a_8\gamma_{\text{degrees}} + a_9F_z + a_{10}\\ &\quad S_v = \Big[(a_{11,1}F_z + a_{11,2})\gamma_{\text{degrees}} + a_{12}\Big]F_z + a_{13}\end{aligned}}$$

There are a lot of ways we could stitch them together. This is not the kind of situation where there is one right answer. Instead, in the absence of hard theory or experimental data, we have the freedom to be creative, with the inevitable risk of being wrong. We pick a method that satisfies some simple, intuitive, physical requirements. First, we must put the inputs on the same footing. Ask "what is the value of σ for which $F_x(\sigma, F_z)$ has its maximum, and what is the value of α for which $F_y(\alpha, F_z, \gamma)$ has its maximum?" Call these two values $\bar{\sigma}$ and $\bar{\alpha}$. They are constants for given F_z and γ: characteristics of a particular tyre and car and surface. So, we can finesse the notation and just write $F_x(\sigma)$ and $F_y(\alpha)$. The maxima identify points on the rim or edge of the 'traction circle'. The grip decreases when σ exceeds $\bar{\sigma}$ and when α exceeds $\bar{\alpha}$. Let's illustrate with $F_z = 3.3\text{KN}$, $\gamma = 0$, and the constants from Genta's alleged Ferrari. Once we substitute all that in (and we'll let you check our arithmetic from the data in prior articles), we get

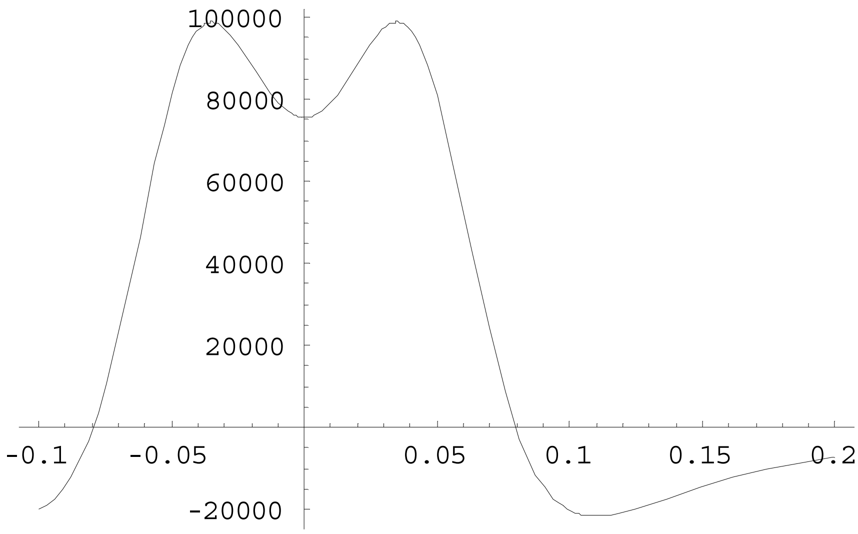
$$\begin{aligned}&F_{x,\,\text{Newtons}}(\sigma) = 5570\sin\Big(1.65\tan^{-1}\Big\{SB - 10\Big[\tan^{-1}(SB) - SB\Big]\Big\}\Big)\\ &\qquad SB = 8.222\,\sigma\end{aligned}$$

$$\begin{aligned}&F_{y,\,\text{Newtons}}(\alpha) = 5570\sin\Big(1.799\tan^{-1}\Big\{SB + E\Big[\tan^{-1}(SB) - SB\Big]\Big\}\Big)\\ &\qquad B = 0.348,\quad E = -0.184,\quad S = \Big[\alpha_{\text{degrees}} - 0.0524\Big]\end{aligned}$$

We evaluate these equations for $\sigma = 0$, $\alpha = 0$, getting $F_x(0) = 0$, $F_y(0) = -71\,\mathrm{N}$, and showing a small lateral force (about 16 lbs) due to conicity and ply steer. The source of that problem is the constant offset in *S*, which results from a_9 and a_{10}'s being non-zero. We just set them to zero for now. Let's plot $F_x(\sigma)$, slip on the horizontal axis and grip on the vertical:

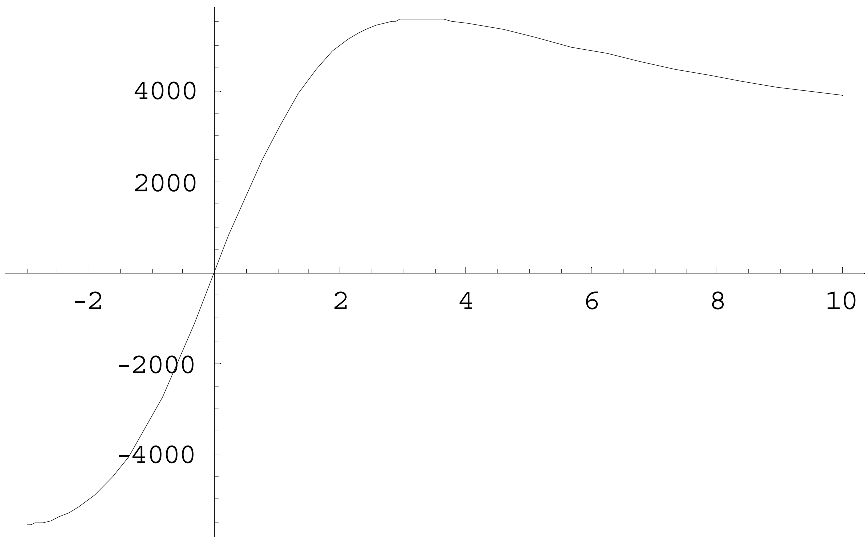


The maximum positive grip occurs, just by eyeball, around $\sigma = 0.08$. To the left of the maximum, adding more slip—more throttle—generates more grip. To the right of the maximum, adding more slip generates *less* grip. That's where we've lost traction. We can find the maximum precisely by plotting the *slope* of this curve, since the slope is zero right at the maximum:

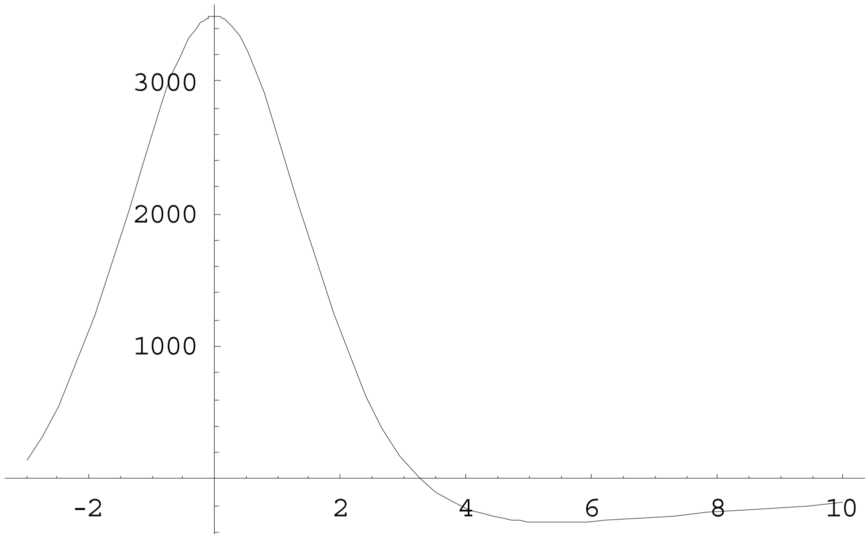


Using secret physicist methods, I've found that this curve crosses the horizontal axis—that is, goes to zero—at precisely $\sigma = 0.0796$.

This was so much fun that we'll just do it again for $F_y(\alpha)$. First, the curve proper:



Notice the same kind of stability situation as we saw before. To the left of the maximum, more slip—more steering—means more grip. To the right of the maximum, more slip means less grip. Here's the slope:



We find that the maximum of the original curve, the zero-crossing of the slope, occurs at α = 3.273°.

Once we find the maxima, we can create new, non-dimensional quantities by scaling σ and α by these values, namely $s = \sigma/\bar{\sigma}, a = \alpha/\bar{\alpha}$. These are pure numbers, so they're commensurable. They are unity when σ and α have the values of maximum traction in isolation of one another. We can then write new functions $\Phi_x(s) = F_x(\sigma)$ and $\Phi_y(a) = F_y(\alpha)$ which have their maxima at $s = 1$ and $a = 1$.

We seek a vector-valued function $\mathbf{F}^+(s,a)$ of s and a whose longitudinal x component F_x^+ expresses the longitudinal force component and whose lateral y component F_y^+ expresses the lateral force component under combination slip. Build this up from Φ_x and Φ_y so that it satisfies the following requirements:

The magnitude of $\mathbf{F}^+$, that is, $\left\|\mathbf{F}^+\right\| \equiv F^+ \equiv +\sqrt{\left(F_x^+\right)^2 + \left(F_y^+\right)^2}$, should have its maximum all the way around the traction circle, that is, whenever $+\sqrt{s^2 + a^2} = 1$.

The individual components should agree completely with the old magic formulae whenever there is pure longitudinal or pure lateral slip. Mathematically, this means that $F_x^+(s,0) = F_x(\sigma) = F_x(s\bar{\sigma})$ and $F_y^+(0,a) = F_y(\alpha) = F_y(a\bar{\alpha})$.

For a fixed, positive value of σ (throttle), as α (steering) increases, the input to F_x must *increase*. Say *what*? Here's the idea. Suppose you're on the limit of longitudinal grip. When steering increases, the forward grip limit must be exceeded, and a great way to model that is just to shove the input over the cliff to larger σ. We want the same behaviour the other way, namely, for a fixed value of α (steering), as σ (throttle) increases, the input to F_y increases to model the fact that at maximum steering adding throttle exceeds the limit. We model the three other cases entailing negative values of σ and α below.

Below the limits, we do not want dramatic increases in forward grip when steering increases, and vice versa. So, although we must increase the input to F_x with increasing α, we must *decrease* the output of F_x. Likewise, while we increase the input to F_y with increasing σ, we must decrease the output. This requirement is a bit of a balancing act because often there *is* an increase of steering grip with braking, as we see in the technique of trail braking. However, there is usually no increase in steering grip with increased throttle in a front-wheel-drive car, even below the limits. In the modelling of combined effects like this, it's necessary to include weight transfer with the combination grip formula. That simply means that until we have a full model of the car up and running, we won't be able to evaluate fully the quality of this combination magic grip formula.

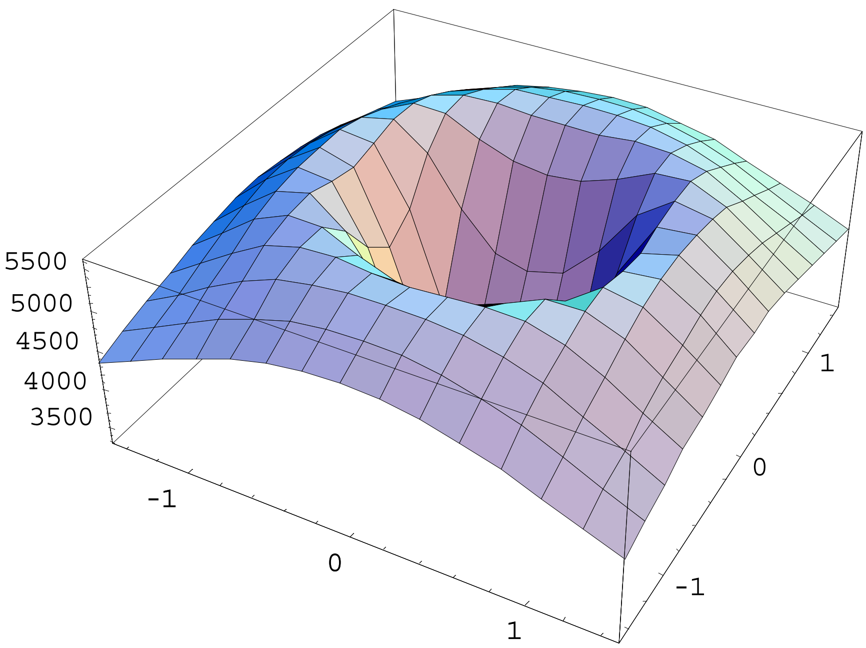
The following table fleshes out requirement 3 for the cases of braking ($\sigma < 0$) or turning left ($\alpha < 0$). The essential idea is that if the magnitude of either parameter increases, then the magnitudes of the inputs to the old magic formulae must increase, but honouring the algebraic signs. If a parameter is positive, it should get more positive as the magnitude of the other parameter increases. Similarly, if a parameter is negative, it should get more negative as the magnitude of the other parameter increases.

$\mathrm{sgn}(\sigma)$	$\mathrm{sgn}(\alpha)$	σ Trend	α Trend	input to F_x	input to F_y
+	+	increasing	fixed	increasing	increasing
+	+	fixed	increasing	increasing	increasing
+	–	increasing	fixed	increasing	decreasing
+	–	fixed	decreasing	increasing	decreasing
–	+	decreasing	fixed	decreasing	increasing
–	+	fixed	increasing	decreasing	increasing
–	–	decreasing	fixed	decreasing	decreasing
–	–	fixed	decreasing	decreasing	decreasing

Without further ado, here's our proposal for the combination magic grip formula:

$$\boxed{\rho = +\sqrt{s^2 + a^2}\ ,\ F_x^+(s,a) = \frac{s}{\rho}\Phi_x(\rho),\ F_y^+(s,a) = \frac{a}{\rho}\Phi_y(\rho)}$$

Using ρ as the input, with the appropriate algebraic signs, satisfies requirements 1. Multiplying the outputs by the ratio of s to ρ and a to ρ magically satisfies requirements 2, 3, and 4. There is, in fact, plenty of freedom in the choice of the outer multiplier: strictly speaking, any power of the ratios would do for requirements 2 and 4, and some care will be required to get the signs right for requirement 3. Until we have a good reason to change it, we'll just go with the ratio straight up, especially since it automatically gets the signs right. We close this instalment with a plot of the magnitude $+\sqrt{\left(F_x^+\right)^2 + \left(F_y^+\right)^2}$ showing the traction circle very clearly:



The stability criteria are visually obvious, here. If the current, commensurable slip values, s and a, are inside the central "cup" region, then increasing either component of slip increases grip. If they're outside, then increasing slip leads to decreasing grip and the driver is in the "deep kimchee" region of the plot.

ERRATA: The *Physics of Racing* series has been fairly error-free over the years, but I caught three small errors in part 22 whilst going over it for this instalment. The good news is that they did not affect any final results. I defined the WHEEL frame at the wheel hub but later I implied that it is centred at the contact patch (CP). In fact, the frame at the CP is the important one, and we call it TYRE from now on, avoiding the ambiguous "WHEEL". We never actually used the improperly defined WHEEL frame, so, again, final results were not affected. Also, the dimensions for a_3 in Part 22 should be N/Degree, not just N, because a_3 furnishes the dimensions for *B*, which always appears in the combination *SB*, and *S* has dimensions of degrees. Finally, the dimensions for a_6 are 1/KN, not KN.

The Physics of Racing, Part 26: The Driving Wheel, Chapter I

Brian Beckman, PhD

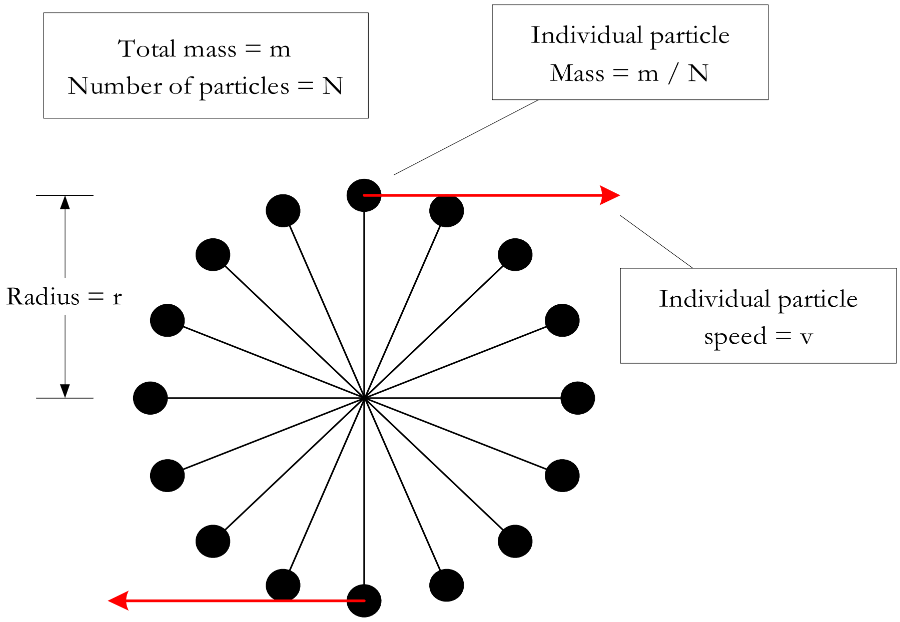
Imagine 400 ft-lbs of torque measured on a chassis dynamometer like a DynoJet (see references at the end). This is a very nice number to have in any car, street or racing. In dyno-speak, however, the interpretation of this number is a little tricky.

To start a dyno session, you strap your car down with the driving wheels over a big, heavy drum, then you run up the gears gently and smoothly until you're in fourth at the lowest usable engine RPM, then you floor it, let the engine run all the way to redline, then shut down. The dyno continuously measures the time and speed of the drum and the engine RPM through a little remote radio receiver that picks up spark-plug noise. The only things resisting the motion of the driving wheels are the inertia of the driveline in the car and the inertia of the drum. The dyno 'knows' the latter, but not the former. Without these inertias loading the engine, it would run up very quickly and probably blow up. Test-stand dynos, which run engines out of the car, load them in different ways to prevent them from free running to annihilation. Some systems use water resistance, others use electromagnetic; in any case, the resistance must be easy to calibrate and measure. We are only concerned with chassis dynos in this article, however.

What is the equation of motion for the car + dyno system? It is a simple variation on the theme of the old, familiar second law of Newton. For linear motion, that law has the form $F = ma$, where F is the net force on an object, m is its mass, and a is its acceleration, or time rate of change of velocity.

For rotational motion, like that of the driveline and dyno, Newton's second law takes the form $T = J\dot{\omega}$ where T is the net torque on an object, J is its ***moment of inertia***, and $\dot{\omega}$ is its ***angular acceleration***, or time rate of change of ***angular velocity***. The purpose of this instalment of the Physics of Racing is to explain everything here and to run a few numbers.

To get the rotational equation of motion, we assume that the dyno drum is strong enough that it will never fly apart, no matter how fast it spins. We model it, therefore, as a bunch of point masses held to the centre of rotation by infinitely strong, massless cables. With enough point masses, we can approximate the smooth (but grippy) surface of the dyno drum as closely as we would like.



Assume each particle receives a force of F/N in the tangential direction. Tangential, of course, means the same as circumferential or longitudinal, as clarified in recent instalments about slip and grip of tires. So, each particle accelerates according to $F/N = ma/N$. The N cancels out, leaving $a = F/m$. Now, a is the rate of change of the velocity, and the velocity is defined as $v = r\omega$, where r is the constant radius of the circle and ω is the angular velocity in ***radians*** per second. The circumference of the circle is $2\pi r$, by definition, so a drum of 3-foot radius has a circumference of about 6.28x3=18.8 feet. At 60 RPM, which is one rev per second, each particle goes 18.8 feet per second, which is about 15x18.8/22=12.8 mph. RPM is one measurement of angular velocity, but it's more convenient to measure it such that 2π angular units go by every second. Such units save us from having to track factors of 2π all over the math. So, there are 2π radians per revolution, and the equation $v = r\omega$ is seen as a general expression of the example that $2\pi \times$ revolutions per second = 18.8 feet per second = velocity.

Since r is constant, it has no rate of change. Only ω has one, measured in radians per second per second, or radians per second squared, or rad/sec^2, and denoted with an overdot: $\dot{\omega}$. The equation of motion, so far, looks like $a = \dot{v} = \dot{\omega}r = F/m$. Now, we know that torque is just force times the lever arm over which the force is applied. So, a force of F at the surface of the drum translates into a torque of $T = Fr$ applied to the shaft—or by the shaft, depending on point of view. So we write $\dot{\omega}r = F/m = T/rm$, which we can rearrange to $\dot{\omega}mr^2 = T$. We can make this *resemble* the linear form of Newton's law if we *define* $J = mr^2$, the moment of inertia of the drum, yielding $T = J\dot{\omega}$, which looks just like $F = ma$ if we analogize as follows: $F \leftrightarrow T$, $J \leftrightarrow m$, $a = \dot{v} \leftrightarrow \dot{\omega}$.

This value for J only works for this *particular* model of the drum, with all the mass elements at distance r from the center. Suffice it to say that a moment of inertia for any other model of the drum could be computed in like manner. It turns out that the moment of inertia of a solid cylindrical drum is half as much, namely $J = \frac{1}{2}mr^2$.

Moments of inertia for common shapes can be looked up all over the place, for instance at http://www.physics.uoguelph.ca/tutorials/torque/Q.torque.inertia.html.

So, now, the dyno has a known, fixed value for *J*, and it measures $\dot{\omega}$ very accurately. This enables it to calculate trivially just how much torque is being applied by the driving wheels of the car to the drum. But it does *not* know the moment of inertia of the driveline of the car, let alone the radius of the wheels, the gear selected by the driver, the final-drive ratio in the differential, and so on. In other words, it knows nothing about the driveline other than engine RPM.

Everyone knows that the transmission and final-drive on a car *multiply* the engine torque. The torque at the driving wheels is almost always much larger than the flywheel torque, and it's larger in lower gears than in higher gears. So, if you run up the dyno in third gear, it will accelerate faster than if you run it up in fourth gear. Yet, the dyno reports will be comparable. Somehow, without knowing any details about the car, not even drastic things like gear choice, the dyno can figure out flywheel torque. Well, yes and no.

It turns out that all the dyno needs to know is engine RPM. It does not matter whether the dyno is run up quickly with a relatively large drive-wheel torque (***DWT***) or run up slowly with a relatively small DWT. Furthermore, the radius of the driving wheels and tires also does not matter. Here's why.

Wheel RPM is directly proportional to drum RPM, assuming the longitudinal slip of the tires is within a small range. The reason is that at the point of contact, the drum and wheel have the same circumferential (longitudinal, tangential) speed, so $v_{\text{wheel}} = r_{\text{wheel}}\omega_{\text{wheel}} = v_{\text{drum}} = r_{\text{drum}}\omega_{\text{drum}}$. Let's write $\omega_{\text{wheel}} = A\omega_{\text{drum}}$, where $A = r_{\text{drum}}/r_{\text{wheel}}$. Engine RPM is related to wheel RPM by a factor that depends on the final-drive gear ratio f and the selected gear ratio $g_i, i = 1,2,3,\ldots$. We write $\omega_{\text{engine}} = B(f, g_i)\omega_{\text{wheel}}$. Usually, engine RPM is much larger than wheel RPM, so we can expect $B(f, g_i)$ to be larger than 1 most of the time. So, we get

$$\omega_{\text{drum}} = \frac{\omega_{\text{wheel}}}{A} = \frac{\omega_{\text{engine}}}{AB(f, g_i)}$$

We also know that, by Newton's Third Law, that the force applied to the drum by the tire is the same as the force applied to the tire by the drum. Therefore the torques applied are in proportion to the radii of the wheel+tyre and the drum, namely that

$$F_{\text{wheel}} = \frac{T_{\text{wheel}}}{r_{\text{wheel}}} = F_{\text{drum}} = \frac{T_{\text{drum}}}{r_{\text{drum}}}$$

or $T_{\text{drum}} = AT_{\text{wheel}}$. Recalling that the transmission gear and final drive multiply engine torque, we also know that $T_{\text{wheel}} = B(f, g_i)T_{\text{engine}}$, so $T_{\text{drum}} = AB(f, g_i)T_{\text{engine}}$. But we already know $AB(f, g_i)$: it's the ratio of the RPMs, so $T_{\text{drum}} = \frac{\omega_{\text{engine}}}{\omega_{\text{drum}}}T_{\text{engine}}$, or, more usefully,

$$T_{\text{engine}} = \frac{\omega_{\text{drum}}}{\omega_{\text{engine}}} T_{\text{drum}} = \frac{\omega_{\text{drum}}}{\omega_{\text{engine}}} J \dot{\omega}_{\text{drum}}$$

Every term on the right-hand side of this equation is measured or known by the dyno, so we can measure engine torque independently of car details! We can even plot T_{engine} versus ω_{engine}, effectively taking the run-up time and the drum data out of the report.

Almost. There is a small gotcha. The engine applies torque indirectly to the drum, spinning it up. But the engine is *also* spinning up the clutch, transmission, drive shaft, differential, axles, and wheels, which, all together, have an unknown moment of inertia that varies from car-to-car, though it's usually considerably smaller than J, the moment of inertia of the drum. But, in the equations of motion, above, we have not accounted for them. More properly, we should write

$$T_{\text{engine}} = \frac{\omega_{\text{drum}}}{\omega_{\text{engine}}} \left(J + J_{\text{miscellaneous}} \right) \dot{\omega}_{\text{drum}}$$

This doesn't help us much because we don't know $J_{\text{miscellaneous}}$, so we pull a fast one and rearrange the equation:

$$\text{define } T_{\text{loss}} = \frac{\omega_{\text{drum}}}{\omega_{\text{engine}}} J_{\text{miscellaneous}} \dot{\omega}_{\text{drum}}$$

$$T_{\text{engine}} - T_{\text{loss}} = \frac{\omega_{\text{drum}}}{\omega_{\text{engine}}} J \dot{\omega}_{\text{drum}}$$

This is why chassis dyno numbers are always lower than test-stand dyno numbers for the same engine. The chassis dyno measures $T_{\text{engine}} - T_{\text{loss}}$, and the test-stand measures T_{engine}. Of course, those trying to sell engines often report the best-sounding numbers: the test-stand numbers. So, don't be disappointed when you take your hot, new engine to the chassis dyno after installation and get numbers 15% to 20% lower than the advertised 'at the crankshaft' numbers in the brochure. It's to be expected. Typically, however, you simply do not know T_{loss}: it's a number you take on faith.

Let's run a quick sample. The following numbers are pulled out of thin air, so don't hang me on them. Suppose the drum has 3-foot radius, is solid, and weighs 6,400 lbs, which is about 200 slugs (remember slugs? One slug of mass weighs about 32 pounds at the Earth's surface). So, the moment of inertia of the drum is about $\frac{1}{2} mr^2 = 900 \text{ slug-ft}^2$. Let's say that the engine takes about 15 seconds to run from 1,500 RPM to 6,000 RPM in fourth gear, with a time profile like the following:

t	e RPM	V MPH	v FPS	drum ω	drum $\dot{\omega}$	drum RPM	RPM ratio	Torque
0	1,500	35	51.33	17.11	0.00	163.40	0.1089	0.00
1	1,800	42	61.60	20.53	3.42	196.08	0.1089	335.51
2	2,100	49	71.87	23.96	3.42	228.76	0.1089	335.51
3	2,400	56	82.13	27.38	3.42	261.44	0.1089	335.51
4	2,700	63	92.40	30.80	3.42	294.12	0.1089	335.51
5	3,000	70	102.67	34.22	3.42	326.80	0.1089	335.51
6	3,300	77	112.93	37.64	3.42	359.48	0.1089	335.51
7	3,600	84	123.20	41.07	3.42	392.16	0.1089	335.51
8	3,900	91	133.47	44.49	3.42	424.84	0.1089	335.51
9	4,200	98	143.73	47.91	3.42	457.52	0.1089	335.51
10	4,500	105	154.00	51.33	3.42	490.20	0.1089	335.51
11	4,800	112	164.27	54.76	3.42	522.88	0.1089	335.51
12	5,100	119	174.53	58.18	3.42	555.56	0.1089	335.51
13	5,400	126	184.80	61.60	3.42	588.24	0.1089	335.51
14	5,700	133	195.07	65.02	3.42	620.92	0.1089	335.51
15	6,000	140	205.33	68.44	3.42	653.60	0.1089	335.51

The "v MPH" column is just a straight linear ramp from 35 MPH to 140 MPH, which are approximately right in my Corvette. The "v FPS" column is just 22/15 the v MPH. The drum ω is in radians per second and is just v FPS divided by 3 ft, the drum radius. The drum $\dot{\omega}$ is just the stepwise difference of the drum ω numbers. It's constant, as we would expect from a run-up of the dyno at constant acceleration. The drum RPM is $60/2\pi$ times the drum ω. The RPM ratio is just drum RPM divided by engine RPM, and it must be strictly constant, so this is a nice sanity check on our math. Finally, the torque column is the RPM ratio times $J = 900$ slug – ft^2 times drum $\dot{\omega}$. We see a constant torque output of about 335 ft-lbs. Not bad. It implies a test-stand number of between 394 and 418, corresponding to 15% and 20% driveline loss, respectively. Looks like we nailed it without 'cooking the books' too badly. Of course, we have a totally flat torque curve in this little sample, but that's only because we have a completely smooth ramp-up of velocity.

Dyno reports often will be labelled 'Rear-wheel torque' (RWT) or, less prejudicially, 'drive-wheel torque' (DWT) to remind the user that there is an unknown component to the measurement. These are well-intentioned *misnomers*: do not be mislead! What they mean is 'engine torque as if the engine were connected to the drive wheels by a massless driveline', or 'engine torque as measured at the drive wheels with an unknown but relatively small inertial loss component'. It should be clear from the above that the actual drive-wheel torque cannot be measured without knowing A, the ratio of the drum radius to the wheel+tyre radius. It's slightly ironic that an attempt to clear up the confusion risks introducing more confusion.

In the next instalment, we relate the equations of motion for the driving wheel to the longitudinal magic formula to compute reaction forces and get equations of motion for the whole car.

References:

http://www.c5-corvette.com/DynoJet_Theroy.htm [sic]

http://www.mustangdyne.com/pdfs/7K%20manualv238.pdf

http://www.revsearch.com/dynamometer/torque_vs_horsepower.html

Attachments:

I've included the little spreadsheet I used to simulate the dyno run. It can be downloaded here.

ERRATA:

* Part 14, yet again, the numbers for frequency are actually in radians per second, not in cycles per second. There are 2π cycles per radian, so the 4 Hz natural suspension frequency I calculated and then tried to rationalize was really 4/6.28 Hz, which is quite reasonable and not requiring any rationalization. Oh, what tangled webs we weave…

* Physical interpretations of slip on page 2 of part 24: "Car (hub) moving forward, CP moving slowly forward w.r.t. ground, resisting car motion." Should be "Car (hub) moving forward, CP moving slowly forward w.r.t. ***HUB***, resisting car motion."

* Part 21, in the back-of-the-envelope numerical calculation just before the 3-D plot at the end of the paper, I correctly calculated $\tan^{-1}(0.822) = 0.688$, but then incorrectly calculated $\tan^{-1}(SB) - SB$ as -0.266. Of course, it's $0.688 - 0.822 = -0.134$. One of the hazards of doing math in one's head all the time is the occasional slip up. Normally, I check results with a calculator just to be really sure, but some are so trivial it just seems unnecessary. Naturally, those are the ones that bite me.

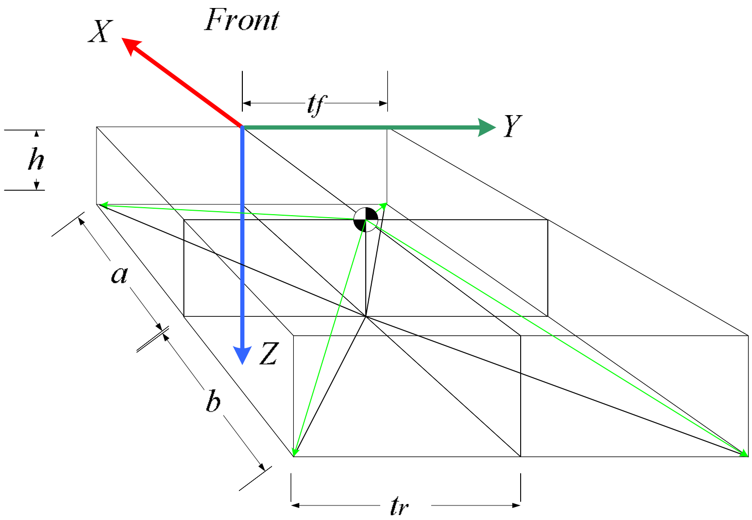
Physics of Racing, Part 27:

Four-Wheel Weight Transfer

Brian Beckman, PhD

In this installment, we revisit the four-wheel statics of Part 20, solving the statics problem for ***level ground***, which is very common in simulation. The problem is: $\boxed{\text{given lateral and longitudinal forces, find the balancing vertical forces}}$. In so doing, we introduce a conventional coordinate system and a new tool: *Mathematica*. This is a comprehensive mathematics package that we use for symbolic manipulation.

First, let's introduce the standard coordinate frame used by the SAE, which is documented in the usual source books by Milliken (*Race Car Vehicle Dynamics*) and Gillespie (*Fundamentals of Vehicle Dynamics*). In this frame, X is forward (***longitudinal***), Y is to driver's right (***lateral***), and Z is downward (***vertical***), in the direction of gravitation. The following figure illustrates:



The symbols have the following meanings:

h	vertical distance of the Center of Gravity (CG) from the ground
a, b	longitudinal distance from the CG to the front and rear axle geometry centers
t_f, t_r	front and rear half-track lateral distances from axle centers to contact patch (CP)

These give us the geometry needed to locate the CPs. Let's code this up in *Mathematica* (MMA, see www.wolfram.com). We open up an MMA 'Notebook' and write

```
tireLocs = {{a, tf, h}, {-b, tr, h},
    {-b, -tr, h}, {a, -tf, h}};
```

This code defines **tireLocs** as a list of 3-vectors, each being a list of locations in three-dimensional space in the car-fixed, SAE coordinate frame. Numbered the tires clockwise from right front (RF). So the RF tire has location $X = a$, $Y = t_f$, and $Z = h$, and so on for the RR, LR, and LF, in order. Note that a, t_f, and h do not need predefined values: MMA treats any undefined symbol as just a symbolic constant (nice!). This is the distinguishing aspect of a symbolic math language like MMA as opposed to an ordinary computer language like C++.

Define a numerical sample early so that it's handy for intuitive checks. The following are for a Lamborghini Diablo, found from a web search, in Meters and Newtons:

```
numRz = {
    a → 1.425, b → 1.029,
    tf → 1.735 / 2, tr → 1.760 / 2,
    h → 0.420, mg → 16680};
```

This code defines a variable, **numRz**, whose value is a list of MMA ***rules***. The following function of two parameters shows how to apply rules:

```
nml8[term_, fRzInput_] := ((term /. numRz) /. fRzInput)
```

This code defines a ***function*** named **nml8** that takes any **term** and applies to it first the rules **numRz** and then any other set of rules, locally named **fRzInput**. The underscores after the names **term** and **fRzInput** in the parameter list are part of MMA's deep pattern-matching syntax. For present purposes, just think of them as necessary noise when defining a function. Test what we have so far as follows:

```
nml8[tireLocs, {}]
{{1.425 , 0.8675 , 0.42 }, {-1.029 , 0.88 , 0.42 },
 {-1.029 , -0.88 , 0.42 }, {1.425 , -0.8675 , 0.42 }}
```

Here is see an example of input and output syntax from MMA. We applied the function **nml8** to the preexisting list of geometry vectors and to a null list of extra rules to get numerical locations of the CPs. These numbers are sensible, by inspection.

Now make a couple of symbolic lists of forces operating on the tires. Separate the forces operating in the $X - Y$ plane from the vertical forces since the former are given and the latter are the final objects of our solution efforts.

```
fxy = {
    {f1x, f1y}, {f2x, f2y}, (* right side *)
    {f3x, f3y}, {f4x, f4y}} (* left side *);
fz = {f1z, f2z, f3z, f4z};
```

For immediate purposes, combine them into three-forces, and there is some magic juju for doing that in MMA:

```
threeForces = MapThread[Append, {fxy, -fz}]
{{f1x , f1y , -f1z }, {f2x , f2y , -f2z }, {f3x , f3y , -f3z }, {f4x , f4y , -f4z }}
```

Lisp programmers will say "Aha!" **MapThread** runs a function, in this case, **Append**, down some lists, and **Append** glues lists together. The two threaded lists are **fxy** and **fz**, defined immediately above. Use the negative of **fz** so that the vertical force vectors point upwards and the force components can be positive numbers. This code makes a new list of four 3-forces named **threeForces**.

Now to the physics. Remember that torque is $\text{lever-arm} \times \text{force}$, where $\times$ is the ***vector cross product***. We have lever-arms in one list, **tireLocs**, and forces in another, **threeForces**, so the torques about the CG at each CP are immediately available:

```
threeTorques = MapThread[Cross, {tireLocs, threeForces}]
{{-f1y h - f1z tf, a f1z + f1x h, a f1y - f1x tf},
 {-f2y h - f2z tr, -b f2z + f2x h, -b f2y - f2x tr},
 {-f3y h + f3z tr, -b f3z + f3x h, -b f3y + f3x tr},
 {-f4y h + f4z tf, a f4z + f4x h, a f4y + f4x tf}}
```

When the car is not flipping over, the X and Y torques are in equilibrium. The Z torque accounts for yaw, so it's free. Add up the X and Y torques:

```
sumTorques = Simplify[Plus @@ threeTorques]
{-f1y h - f2y h - f3y h - f4y h - f1z tf + f4z tf - f2z tr + f3z tr,
 -b (f2z + f3z) + a (f1z + f4z) + (f1x + f2x + f3x + f4x) h,
 -b (f2y + f3y) + a (f1y + f4y) - f1x tf + f4x tf - f2x tr + f3x tr}
```

"@@" is MMA juju for applying the function **Plus** across a list of vectors. It's similar to **MapThread**, but not quite the same (see the MMA documentation for details). Construct and solve the X and Y equations for torque equilibrium:

```
torqueEquations = {sumTorques[[1]] == 0, sumTorques[[2]] == 0};
```

The double square brackets pick out elements of lists, so **sumTorques[[1]]** is the first element of sumTorques, that is, the torque about the X axis. The double-equals is an assertion that **sumTorques[[1]]** is zero, and solvable MMA equations must contain double equals. We have a similar equation for **sumTorques[[2]]**, the Y torque. Thus, **torqueEquations** is a list of two equations. Solving:

```
rightHandRules = Solve[torqueEquations, {f1z, f2z}] // FullSimplify
```

$$\left\{\left\{\text{f1z} \to -\frac{(\text{a f4z} + (\text{f1x} + \text{f2x} + \text{f3x} + \text{f4x})\,\text{h})\,\text{tr} + \text{b}\,((\text{f1y} + \text{f2y} + \text{f3y} + \text{f4y})\,\text{h} - \text{f4z tf} - 2\,\text{f3z tr})}{\text{b tf} + \text{a tr}},\right.\right.$$
$$\left.\left.\text{f2z} \to \frac{-\text{b f3z tf} + (\text{f1x} + \text{f2x} + \text{f3x} + \text{f4x})\,\text{h tf} - \text{a}\,((\text{f1y} + \text{f2y} + \text{f3y} + \text{f4y})\,\text{h} - 2\,\text{f4z tf} - \text{f3z tr})}{\text{b tf} + \text{a tr}}\right\}\right\}$$

This code solves the equations for the specified variables, **f1z** and **f2z**, expressing the solution as rules that can be applied in other contexts. We've already seen numerical rules in action, but we can have symbolic ones too, and equation solutions are an example.

These solutions are a 'mouthful', but notice, slightly surprisingly, that the lateral and longitudinal forces show up only as their sums, so reduce the amount of 'verbiage' by defining and applying a couple of rules by hand

```
fxRule = f1x + f2x + f3x + f4x → fx;
fyRule = f1y + f2y + f3y + f4y → fy;
rhr = rightHandRules /. {fxRule, fyRule}
```

$$\left\{\left\{\text{f1z} \to -\frac{(\text{a f4z} + \text{fx h})\ \text{tr} + \text{b}\ (\text{fy h} - \text{f4z tf} - 2\ \text{f3z tr})}{\text{b tf} + \text{a tr}},\ \text{f2z} \to \frac{-\text{b f3z tf} + \text{fx h tf} - \text{a}\ (\text{fy h} - 2\ \text{f4z tf} - \text{f3z tr})}{\text{b tf} + \text{a tr}}\right\}\right\}$$

```
lhr = leftHandRules /. {fxRule, fyRule}
```

$$\left\{\left\{\text{f4z} \to \frac{-(\text{a f1z} + \text{fx h})\ \text{tr} + \text{b}\ (\text{fy h} + \text{f1z tf} + 2\ \text{f2z tr})}{\text{b tf} + \text{a tr}},\ \text{f3z} \to \frac{(-\text{b f2z} + \text{fx h})\ \text{tf} + \text{a}\ (\text{fy h} + 2\ \text{f1z tf} + \text{f2z tr})}{\text{b tf} + \text{a tr}}\right\}\right\}$$

Notice the solution for the left-hand side (LHS) of the car, obtained by methods analogous to those for the right-hand side (RHS). These expressions are much more digestible. The first thing to notice is that the solutions for **f1z** and **f2z**, on the RHS, depend on the solutions for **f3z** and **f4z** on the LHS. This is no help. As discussed in Part 20 of the *Physics of Racing*, we need more information. Posit that cross ratios of weights are equal: that any weight-jacking in the car is symmetric. For instance, the ratio of left to right is the same at front as at rear, or, equivalently, that the ratio of front to rear is the same on left as at right. These conditions yield another equation.

```
eq1 = f1z f3z == f2z f4z;
```

Get one more equation from force equilibrium: the sum of all vertical loads equals the weight of the car.

```
eq2 = mg == (f1z + f2z + f3z + f4z);
```

Solve for rules to eliminate **f3z** and **f4z** from the right-hand rules obtained above:

```
s34 = Solve[{eq1, eq2}, {f3z, f4z}]
```

$$\left\{\left\{\text{f3z} \to -\frac{\text{f2z}\ (\text{f1z} + \text{f2z} - \text{mg})}{\text{f1z} + \text{f2z}},\ \text{f4z} \to -\frac{\text{f1z}\ (\text{f1z} + \text{f2z} - \text{mg})}{\text{f1z} + \text{f2z}}\right\}\right\}$$

```
rTea = Flatten[FullSimplify[rhr /. s34]]
```

$$\left\{\text{f1z} \to -\frac{1}{\text{b tf} + \text{a tr}}\left(\left(\text{fx h} + \text{a f1z}\left(-1 + \frac{\text{mg}}{\text{f1z} + \text{f2z}}\right)\right)\text{tr} + \text{b}\left(\text{fy h} + \frac{(\text{f1z} + \text{f2z} - \text{mg})\ (\text{f1z tf} + 2\ \text{f2z tr})}{\text{f1z} + \text{f2z}}\right)\right),\ \text{f2z} \to \frac{1}{\text{b tf} + \text{a tr}}\left(\text{fx h tf} + \frac{\text{b f2z}\ (\text{f1z} + \text{f2z} - \text{mg})\ \text{tf}}{\text{f1z} + \text{f2z}} - \text{a}\left(\text{fy h} + \frac{(\text{f1z} + \text{f2z} - \text{mg})\ (2\ \text{f1z tf} + \text{f2z tr})}{\text{f1z} + \text{f2z}}\right)\right)\right\}$$

The important thing here is the expression **rhr/.s34**, which applies the elimination rules to **rhr**. The functions **FullSimplify** and **Flatten** are afterthoughts to reduce the verbosity of the symbolic results.

This is great. We have expressions that depend *only* on f1z and f2z, so we have successfully isolated the RHS. Convert these rules to equations thusly, and solve again:

```
rNoah = Map[Apply[Equal, #] &, rTea]
rSoln = Simplify[Solve[rNoah, {f1z, f2z}]]
sf1z = rSoln[[1, 1, 2]];
```

```
sf1zs = FullSimplify[sf1z]
```

$$\frac{1}{2}\left(\frac{-\text{fx h} + \text{b mg}}{\text{a} + \text{b}} + \frac{\text{fy h (fx h} - \text{b mg)}}{\text{fx h (}-\text{tf} + \text{tr)} + \text{mg (b tf} + \text{a tr)}}\right)$$

Some of the above, you must take on faith due to lack of space to explain. Suffice it to say that we do likewise for tires 2, 3, and 4, getting

```
sf2zs = FullSimplify[sf2z]
```

$$\frac{1}{2}\text{(fx h} + \text{a mg)}\left(\frac{1}{\text{a} + \text{b}} - \frac{\text{fy h}}{\text{fx h (}-\text{tf} + \text{tr)} + \text{mg (b tf} + \text{a tr)}}\right)$$

```
sf4zs = FullSimplify[sf4z]
```

$$\frac{1}{2}\left(\frac{-\text{fx h} + \text{b mg}}{\text{a} + \text{b}} - \frac{\text{fy h (fx h} - \text{b mg)}}{\text{fx h (}-\text{tf} + \text{tr)} + \text{mg (b tf} + \text{a tr)}}\right)$$

```
sf3zs = FullSimplify[sf3z]
```

$$\frac{1}{2}\text{(fx h} + \text{a mg)}\left(\frac{1}{\text{a} + \text{b}} + \frac{\text{fy h}}{-\text{fx h tf} + \text{b mg tf} + \text{fx h tr} + \text{a mg tr}}\right)$$

Finally, we check the results

```
FullSimplify[
 (FullSimplify[sumTorques /. fxRule] /. fyRule) /.
   lSoln /. rSoln]
{{{0, 0, -b (f2y + f3y) +
    a (f1y + f4y) - f1x tf + f4x tf - f2x tr + f3x tr}}}
```

getting 0 for the X and Y torques, as required. Visually, we see regular patterns in the solutions above. returning to traditional notation, first define

$$t_{\Delta F} = (b\,mg - F_x\,h)/2$$

$$t_{\Delta R} = (a\,mg + F_x\,h)/2$$

$$\bar{l} = 1/(a+b)$$

$$\bar{R}_A = \frac{hF_y}{hF_x(t_r - t_f) + mg(at_r + bt_f)}$$

The first two terms have the dimensions of torques, that is, force by distance. It is, however, difficult to interpret them as torques on the chassis with some sort of intuitively meaningful relevance to the problem at hand. Think of them as just some expressions with interesting form extruded from the solution. The next two terms have the dimensions of inverse length. The forces, then, have the following convenient, almost pretty forms:

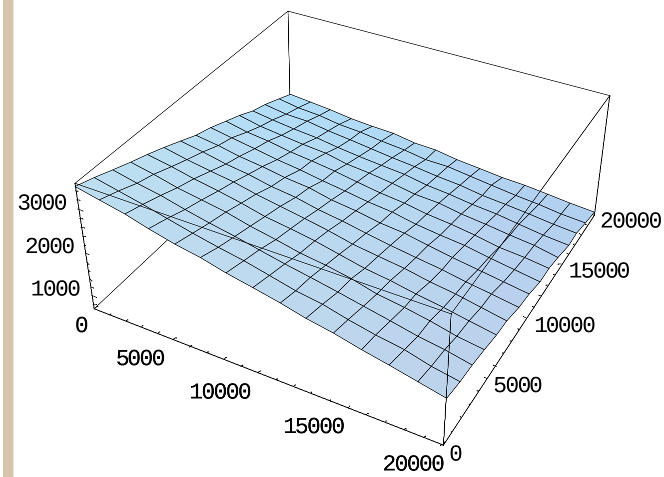
$$F_{[LF]z} = t_{\Delta F}\left(\bar{l} + \bar{R}_A\right) \quad F_{[RF]z} = t_{\Delta F}\left(\bar{l} - \bar{R}_A\right)$$
$$F_{[LR]z} = t_{\Delta R}\left(\bar{l} + \bar{R}_A\right) \quad F_{[RR]z} = t_{\Delta R}\left(\bar{l} - \bar{R}_A\right)$$

Finally, we can use some of MMA's graphics functions to get a visual check on these results. Apply the numerical rules to the solution for tire 1

```
f1zPrototype = Simplify[nml8[sf1z, {}]]
(0.203749 (-17163.7 + 0.42 fx)
   (-35806.2 - 0.00525 fx + 1.03068 fy)) /
 (35806.2 + 0.00525 fx)
```

Redefine it as a function by hand so we can plot it

```
func1z[fx_, fy_] :=
 (0.203749 * (-17163.72 + 0.42 * fx) *
    (-35806.25 - 0.00525 * fx + 1.03068 * fy)) /
  (35806.25 + 0.00525 * fx)
Plot3D[func1z[x, y], {x, 0, 20000}, {y, 0, 20000}]
```



As expected intuitively, the weight on tire 1, the right front, decreases with increasing lateral and longitudinal forces, which range from 0 to 20,000 Newtons in the plot. As the longitudinal force increases, the car is forced forward and the weight is taken off the nose. As the lateral force increases, the car is forced rightwards and weight transfers to the left as in a right-hand turn. The other three tires behave likewise reasonably.

We have posed and solved a familiar racing problem using a programming language for symbolic mathematics. We can code up these solutions in an ordinary language like C++ and use them in our simulation program. This methodology illustrates the application of one programming language, MMA in this case, to the design of software in another programming language. In fact, there is very significant time savings in using powerful tools like this, since the alternative is coding up algebraic mistakes in C++ and then debugging them in the context of a running simulation. This latter approach is very, very time-consuming and labor-intensive. In future articles, we will use another tool, Matlab, to design some more simulation software.

The Physics of Racing, Part 28: Hazards of Integration

Brian Beckman, PhD

The equations of motion are ***differential equations***. Such equations tell us how to calculate "what's happening now" from "what happened a little while ago." They're called *differential* because they have the form of ratios of ***differences*** (*qua* subtraction or ***differentiation*** or ***derivatives***) between "what's happening now" and "what happened a little while ago." The process of ***solving*** the equations, that is, finding out "what's happening now," is ***integration***, a kind of addition, which reverses the subtraction of differentiation. In computer simulation, we integrate numerically, but numerical integration is fraught with hazards. In this article, we give a stark illustration of a very simple differential equation going massively haywire under a very straightforward integration technique. We also introduce MATLAB (from www.mathworks.com) as a programming, visualization, and design tool.

As usual, let's start with Newton's Second Law (***NSL***), $F = ma$. Acceleration, a, is the rate of change or *first derivative* of velocity, which is how much velocity changes over a small interval of time. Consider the following chain of definitions:

$$a(t=t_2) = \lim_{t_2 \to t_1}\left(\frac{v(t_2)-v(t_1)}{t_2-t_1}\right) \stackrel{\text{def}}{=} \lim_{\Delta t \to 0}\left(\frac{\Delta(v(t))}{\Delta t}\right) \stackrel{\text{def}}{=} \frac{d(v(t))}{dt} \stackrel{\text{def}}{=} \frac{dv}{dt}$$

The first term says "acceleration at time $t = t_2$ is the ratio of the difference of the velocities at two times ("now," t_2, and "a little while ago," t_1), divided by the difference of the two times, calculated as we push the two times closer and closer." The velocity differences will get smaller and smaller, but the time differences will get smaller and smaller, too, so the ratio will, one hopes, converge to a certain number, and we call that number the acceleration. If the acceleration is large, the velocity will be changing quickly over short times, so the difference $v(t_2)-v(t_1)$ will be large compared to $t_2 - t_1$. We gloss over lots of detail, here. If you need a refresher, let me suggest looking up "differential calculus" on www.britannica.com.

We can see that "what's happening now," namely $v(t_2)$ depends on "what happened a little while ago," namely $v(t_1)$, and on mass m and force F. Likewise velocity is the rate of change of position, $v = dx/dt$. So, just as the velocity *now* depends on the velocity *a little while ago* through the acceleration, the position *now* depends on the position *a little while ago* through the velocity. We use two, linked, first-order differential equations to get position.

Numerically, as in simulation, we cannot actually collapse the two times. That's only possible in a ***symbolic*** solution, also called ***exact***, ***closed-form***, or ***analytic***. Rather, in a typical simulation setting, Δt, the ***integration step size***, will be set by the environment, often by a graphics rendering loop. At 30 frames per second, an acceptable minimum, Δt will be about 33.3 milliseconds (msec) or 1/30 seconds. At 100 miles per hour, or 147 feet per second, a car will go about $147/30 \approx 4.8$ feet in that time. This means that with an integration step size of 33 msec, we can only predict the car's motion every five feet at typical racing speeds. This back-of-the-envelope calculation should make us a little nervous.

To find out how bad things can get, we need an example that we can solve analytically so we can compare numerical solutions to the exact one. A whole car is far, far too complex to solve analytically, but we usually model many parts of a car as damped harmonic oscillators (***DHO***s), and a DHO can be solved exactly. So this sounds like a highly relevant and useful sample.

In fact, it turns out that we can get into numerical trouble with an *undamped* oscillator, and it doesn't get much simpler than that. So, consider a mass on a spring in one dimension. The location of the mass is $x(t)$, its mass is *m*, the spring constant is *k*, and the mass begins at position $x(t=0)=x_0$ and velocity $v(t=0)=0$. Hooke's Law tells us that the force is $-kx(t)$, so the differential equation looks like

$$a(t)=\frac{dv(t)}{dt}=\frac{d\,\frac{dx(t)}{dt}}{dt}\stackrel{def}{=}\frac{d^2x(t)}{dt^2}=-kx(t);\ x(0)=x_0,\ v(0)=0$$

The process for solving differential equations analytically is a massive topic of mathematical study. We showed one way to go about it in Part 14 of the *Physics of Racing*, but there are hundreds of ways. For this article, we'll just write down the solution and check it.

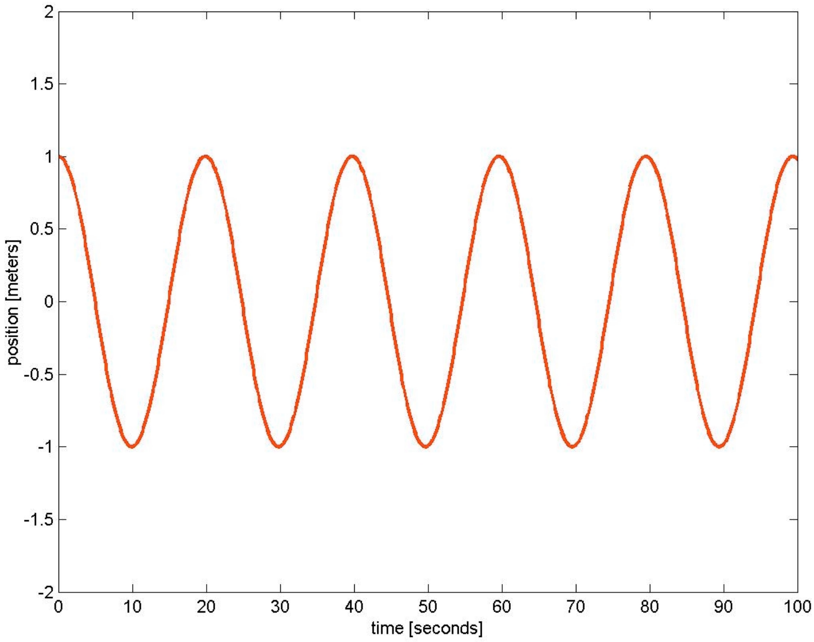
$$x(t)=x_0\cos(\omega t)$$

$$\frac{dx}{dt}=-\omega\sin(\omega t);\ \frac{d^2x}{dt^2}=-\omega^2\cos(\omega t)=a(t)$$

$$m\,a(t)=-k\,x(t)\text{ iff }m\,\omega^2=k\text{ or }\boxed{\omega=\pm\sqrt{k/m}}$$

ω is the ***angular frequency*** in radians per second, or the reciprocal of the amount of time the oscillator's argument takes to complete 1 radian of a cycle. Since there are 2π radians in a cycle, the ***period*** of the oscillator, or the amount of time to complete a cycle, is $2\pi/\omega$ [seconds per cycle = radians per cycle/radians per second].

Let's plot a specific example with numbers picked out of a hat: $m=10\,\text{kg}$, or about 98 Newtons or 22 lbs; $k=1\,N/m$. We expect the period to be $2\pi\sqrt{10}=6.28\times3.17=19.9$ or almost 20.



Sure enough, one cycle of the oscillator takes about 20 seconds. We used MATLAB to generate this plot. This is an interactive programming environment with matrices as first-class objects, meaning that pretty much everything is a matrix. We first set up a 1-dimensional matrix, or vector, of time points

```
h  = .100   ;
tn = 100    ;
tt = 0:h:tn ;
```

Read this as follows: `h` is a (scalar) time step equal to 0.1 seconds; `tn` is an upper bound (scalar) equal to 100 seconds; `tt` is a vector beginning at time 0, ending at time `tn`, and having a value every `h` seconds. `tt` has 1001 elements, namely $0.1, 0.2, 0.3, \ldots, 99.9, 100.0$. Sample the solution, $x_0 \cos(\omega t)$, at these points and plot it with the following commands:

```
plot (tt, cos(tt/sqrt(10)), '-r', 'LineWidth', 2)
axis ([0, tn, -2 2])
xlabel ('time [seconds]')
ylabel ('position [meters]')
```

The string `'-r'` means ‘use a straight red line’. The only other notation that might not be obvious in the above is the argument of `axis`, namely `[0, tn, -2 2]`. This is an explicit or literal vector of limits for the axes on the plot. It’s a little strange that the axis limits and the labels are specified *after* the `plot` command, but that’s the way MATLAB works. Note that the missing comma between -2 and 2 is not a typo: commas are optional in this context.

Now that we know what the exact solution looks like, let’s do a numerical integration. Not long after the differential calculus was invented by Liebniz and Newton, Leonhard Euler articulated a numerical method. It’s the most straightforward one imaginable. Suppose we have a fixed, non-zero time step, Δt. Then we may write an approximate version of the equation of motion as $-k\,x(t) = m\left[v(t+\Delta t) - v(t)\right]/\Delta t$. If we know $x(t)$ and $v(t)$, position and velocity at time t ("a little while ago"), then we can easily solve this equation for the unknown $v(t+\Delta t)$, velocity "now," namely $\left[-k\,x(t)\Delta t/m\right] + v(t)$. Likewise, we approximate the position equation as $\left[x(t+\Delta t) - x(t)/\Delta t\right] = v(t)$, or $x(t+\Delta t) = v(t)\Delta t + x(t)$. With these two equations, all we need is $x(0)$ and $v(0)$ and we can numerically predict the motion forever. Here's the MATLAB code:

```
L  = length(tt) ;
x0 = [1, 0]'    ;
exnp(:,1)  = x0 ;
for i=2:L
    exnp(:,i) = euler('springfunc', tt(i-1), exnp(:,i-1), h) ;
end
```

The first thing to note here is that we've packaged up position and velocity in another vector. This is very convenient since MATLAB prefers vectors and matrices, and it's possible because both equations of motion are first-order by design, that is, they each contain a single, first-derivative expression. We then integrate this pair of equations by calling **`euler`** with a function name in a character string, the time *a little while ago*, the solution vector *a little while ago*, and the integration step size, here written **`h`**. Here's **`euler:`**

```
function [xnp1, tnp1] = euler (f, tn, xn, h)
  fcall1 = [f '(' num2str(tn) ', ' vec2str(xn) ''')'] ;
  k1 = eval (fcall1) ;
  xnp1 = xn + k1 * h ;
  tnp1 = tn + h ;
  return ;
```

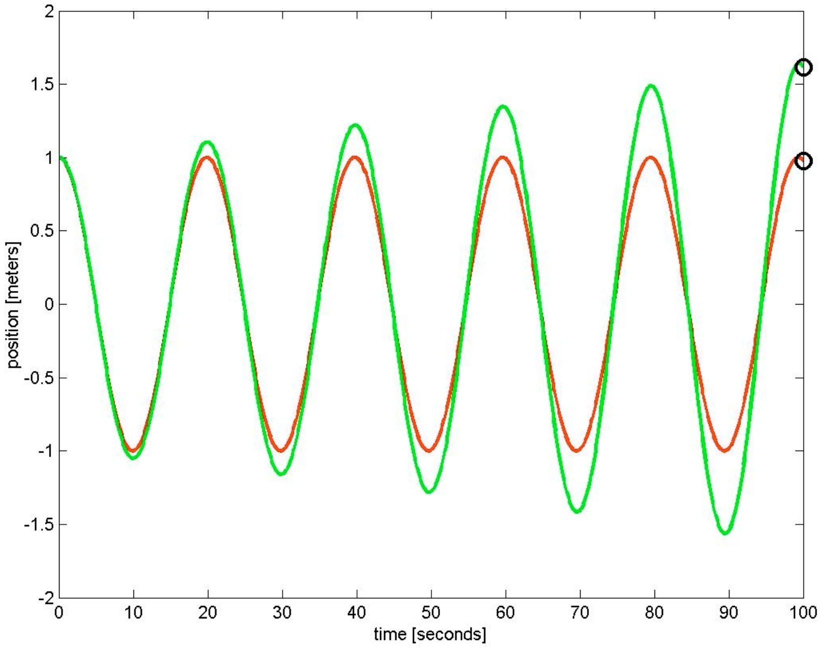
Euler builds up a string from the function name and its arguments, calls the function via the built-in **`eval`** instruction, then updates the solution vector and the time and returns them in a new vector. The **`eval`** instruction is the equivalent of calling a function through a pointer, which should be a familiar concept to C++ programmers. Here's the function we call:

```
function xdot = springfunc(t, x)
    m =   10 ;% kg, about 98 Newtons or 22 lbs
    k =    1 ;% Newton / meter
    matrix = [    0        1
                -k/m       0   ] ;
    xdot = matrix * x ;
```

The function models the pair of differential equations by a matrix multiplication. In traditional notation, here's what it's doing:

$$\begin{pmatrix} \Delta x/\Delta t \\ \Delta v/\Delta t \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -k/m & 0 \end{pmatrix} \begin{pmatrix} x \\ v \end{pmatrix}, \text{ or } \begin{array}{l} \Delta x/\Delta t = v \\ \Delta v/\Delta t = -k\,x/m \end{array}$$

A few moments' thought should convince anyone still with us that the MATLAB code is doing just what we want it to do. Let's plot:



DISASTER! The numerical version is 60% larger than it should be at 100 seconds, and looks as though it will continue to grow without bound. What could be wrong? Let's look back at our approximate equations.

$$v(t+\Delta t) = \left[-k\,x(t)\Delta t/m\right] + v(t)$$
$$x(t+\Delta t) = v(t)\Delta t + x(t)$$

What would happen if a small error, say ε, crept into the velocity? Instead of the above, we would have, at the first step,

$$\tilde{v}(t+\Delta t) = \left[-k\,x(t)\Delta t/m\right] + v(t) + \varepsilon = \boxed{v(t+\Delta t)+\varepsilon}$$
$$\tilde{x}(t+\Delta t) = \left[v(t)+\varepsilon\right]\Delta t + x(t) = \boxed{x(t+\Delta t)+\varepsilon\,\Delta t}$$

The predicted velocity, $\tilde{v}$, will be in error by ε and the position $\tilde{x}$ by $\varepsilon\,\Delta t$. If no further errors creep in, what happens at the next step?

$$
\begin{aligned}
\tilde{v}(t+2\Delta t) &= \left[-k\,\tilde{x}(t+\Delta t)\,\Delta t/m\right] + \tilde{v}(t+\Delta t) \\
&= \left[-k\left\{x(t+\Delta t)+\varepsilon\,\Delta t\right\}\Delta t/m\right] + v(t+\Delta t)+\varepsilon \\
&= \left[-k\,x(t+\Delta t)\,\Delta t/m\right] + v(t+\Delta t) - k\varepsilon\,\Delta t^2/m+\varepsilon \\
&= \boxed{v(t+2\Delta t) - k\,\varepsilon\,\Delta t^2/m+\varepsilon} \\
\tilde{x}(t+2\Delta t) &= \tilde{v}(t+\Delta t)\,\Delta t + \tilde{x}(t+\Delta t) \\
&= \left[v(t+\Delta t)+\varepsilon\right]\Delta t + x(t+\Delta t)+\varepsilon\,\Delta t \\
&= v(t+\Delta t)\,\Delta t + x(t+\Delta t) + 2\varepsilon\,\Delta t \\
&= \boxed{x(t+2\Delta t)+2\varepsilon\,\Delta t}
\end{aligned}
$$

The position error gets WORSE, even when no further errors creep into the velocity. The velocity errors might not get worse as quickly; much depends on the size of $k\,\Delta t^2/m$ relative to ε. However, working through another step, we can conclude that

$$
\begin{aligned}
\tilde{v}(t+N\Delta t) &= v(t+\Delta t) - k\,N\varepsilon\,\Delta t^2/m+\varepsilon \\
\tilde{x}(t+N\Delta t) &= x(t+N\Delta t) + N\varepsilon\,\Delta t
\end{aligned}
$$

so the velocity errors will *eventually* overwhelm $k\,\Delta t^2/m$ as N grows.

What is to be done? The answer is to use a different numerical integration scheme, one that samples more points in the interval and detects curvature in the solution. The fundamental source of error growth in the Euler scheme is that it is a linear approximation: the next value depends linearly on the derivative and the time step. But it is visually obvious that the solution function is curving and that a linear approximation will overshoot the curves.

In this article, to keep it short, we simply state the answer and demonstrate its efficacy. The 4$^{\text{th}}$-order Runge-Kutta method is the virtual industry standard. In a later article, we intend to present an original derivation (done without sources for my own amusement, as is usual in this series), if the length turns out to be reasonable and level of detail remains instructive. This method samples the solution at four points interior to each integration step and combines them in a weighted average. For now, take the location of the interior sample points and the magnitudes of the averaging weights on faith: they're the subject of the upcoming derivation. In the mean time, you can look up various derivations easily on the web. Here's the cookbook recipe: for full generality, rewrite the equation as a derivative's equalling an *arbitrary* function of time and the solution (this is much more general than our specific case):

$$
\frac{dx}{dt} = f(t,x)
$$

Then, chain solution steps to one another as follows:

$$x(t+\Delta t) = x(t) + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4), \text{ where}$$
$$k_1 = f(t, x(t))$$
$$k_2 = f\left(t + \frac{\Delta t}{2}, x(t) + \frac{\Delta t}{2} k_1\right)$$
$$k_3 = f\left(t + \frac{\Delta t}{2}, x(t) + \frac{\Delta t}{2} k_2\right)$$
$$k_4 = f(t + \Delta t, x(t) + \Delta t\, k_3)$$

This is a lot easier to implement than it looks, especially when we rewrite the old Euler scheme in a similar fashion

$$x(t+\Delta t) = x(t) + \Delta t\, k_1, \text{ where } k_1 = f(t, x(t))$$

and when we show the MATLAB code for it:

```
function [xnp1, tnp1] = rk4 (f, tn, xn, h)
  h2 = h/2 ;
  fcall1 = [f '(' num2str(tn) ', ' vec2str(xn) ''')'] ;
  k1 = eval (fcall1) ;
  fcall2 = [f '(' num2str(tn+h2) ', ' vec2str(xn + h2 * k1) ''')'] ;
  k2 = eval (fcall2) ;
  fcall3 = [f '(' num2str(tn+h2) ', ' vec2str(xn + h2 * k2) ''')'] ;
  k3 = eval (fcall3) ;
  fcall4 = [f '(' num2str(tn+h) ', ' vec2str(xn + h * k3) ''')' ];
  k4 = eval (fcall4) ;

  xnp1 = xn + (h/6)*(k1 + 2*k2 + 2*k3 + k4) ;
  tnp1 = tn + h ;

  return ;
```

Make a plot exactly as we did with Euler, in fact, let's plot them both on top of each other in a 'movie' format along with the exact solution (it's only possible to appreciate the animation fun, here, if you have a running copy of MATLAB):

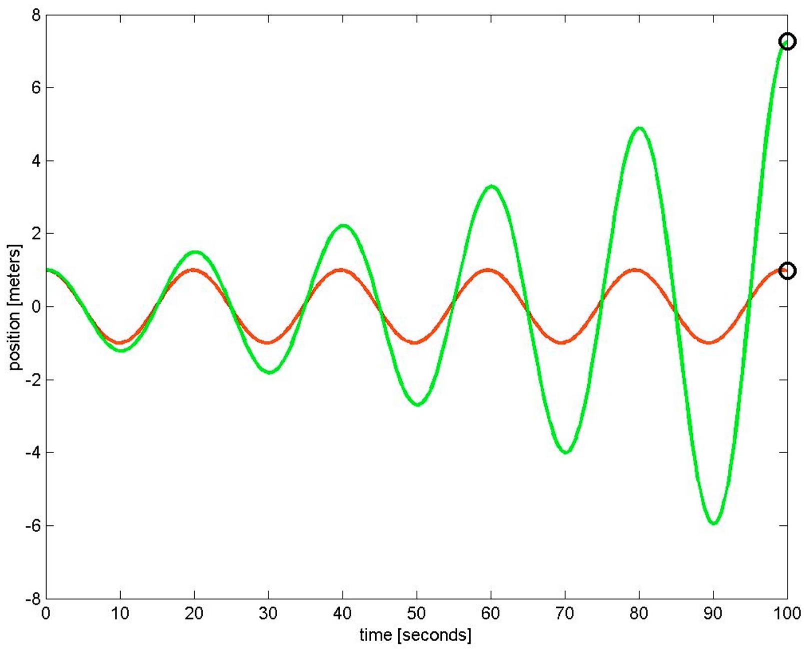
```
fn = figure ;
set (fn, 'DoubleBuffer', 'on');
for i=2:L
    rkxnp(:,i) = rk4  ('springfunc', tt(i-1), rkxnp(:,i-1), h) ;
    exnp (:,i) = euler('springfunc', tt(i-1), exnp (:,i-1), h) ;
    exact(:,i) = cos (tt(i)/sqrt(10));

    plot (tt(1:i), rkxnp(1,1:i), '-b', ...
          tt(1:i), exnp (1,1:i), '-g', ...
          tt(1:i), exact(1,1:i), '-r', ...
          tt(i), exnp (1,i), 'bo', ...
          tt(i), rkxnp(1,i), 'go', ...
          'LineWidth', 2, ...
          'MarkerEdgeColor','w', ...
          'MarkerSize',10);

    axis ([0 tn -2 2]);
    xlabel ('time [seconds]');
```

```
    ylabel ('position [meters]');
    drawnow;
end
```

In the following plot, we show the results of running the code above with $\Delta t = h$ changed to 0.4 from 0.1, showing how the errors in `euler` accumulate much faster over the same time span (we also adjusted the axis limits for the larger errors). The most important thing to notice here is how the Runge-Kutta solution remains completely stable and visually indistinguishable from the exact solution while the Euler method goes completely mad.

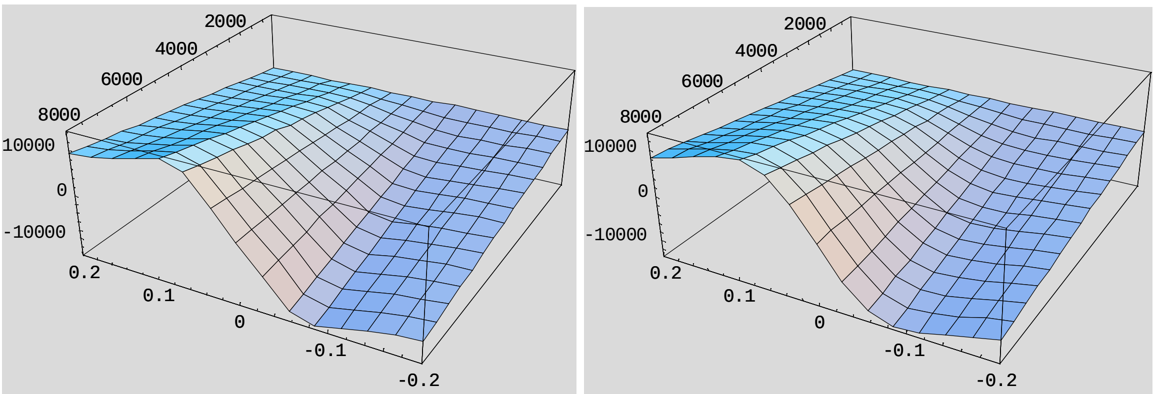


Physics of Racing, Part 29:

A Magical Trick

Brian Beckman, PhD
Copyright[1] Feb 2002

The Magic Formulae (*Physics of Racing* parts 21, 22, 24, 25) for grip *versus* slip have some disadvantages for the simulation programmer. Chief among them are the complicated mathematical structure and the large number of parameters. In this installment, we show a much simpler mathematical expression that mimics the most important, overall features of the Magic Formulae with many fewer parameters. This "Magic Trick" formula is much easier to code and debug, requires much less computing horsepower, and differs by less than 10% almost everywhere from Pajecka: probably sufficiently accurate for gaming simulation. Just to tantalize you, let me offer the following two plots:



The one on the left is the full, longitudinal Magic Formula from Part 21. The one on the right is a plot of $f(s, F_z) = 31\, sF_z \Big/ \left(1 + |9.625\, s|^{2.375}\right)$, where s is longitudinal slip and F_z is vertical load in Newtons, as before. This function is much easier to grasp and remember than is the Pajecka formula. Also, it is plain to see that the two plots are so similar that we should expect no gross handling differences were we to use the right-hand in lieu of the left-hand. In the rest of this installment, I detail the methods for obtaining the numerical values of the three parameters,

The full Magic Formulae were developed for professional research in tire dynamics. That is why they have so many parameters: they must model everything from large truck tires to off-road tires to racing slicks, plus every subtlety of behavior of every different type of tire. For simulation and gaming, however, the most important behavior is simply breakaway instability: the decrease in grip with increasing slip beyond the limits. Minor bumps and wiggles in the curves may be critical to capture differences in brand or type or rubber compound, but they are probably overkill even for detailed simulations like Grand Prix Legends or NASCAR 4. Since the Magic Formulae are "the only game in town," however, simulation programmers tend to use them whilst purposefully wasting their subtle capabilities. In fact, my own presentations of the formulae set most of the parameters to zero value, and mine was derived directly from Genta's (*Motor Vehicle Dynamics*, pub. World Scientific, 1999). So the available modeling 'dynamic range,' as it were, of the Magic Formulae, seems too large for our applications. We need a simpler formula. After playing around for a little while, I found that the following bears an uncanny resemblance to the Magic Formula, but with only three parameters, (A, B, P):

$$F_{\text{horizontal}} = \frac{B\alpha}{1+|A\alpha|^P}$$

This is just a generic form of the formula, in which $F_{\text{horizontal}}$ could be F_{lateral} or $F_{\text{longitudinal}}$, and α could be slip angle or slip ratio. In other words, the sketch above can account for either the longitudinal (part 21) or the lateral Magic Formula (part 22). Let's analyze the longitudinal formula in detail, using Mathematica (MMA, introduced in *PhoRS* part 27). I make reference to the longitudinal formula from part 21 *without* copying it here, so you will need part 21 on hand to go forward from this point. You will also need familiarity with Mathematica, and possibly to have its manual on hand.

Encode the Pajecka b parameters twice as vectors, one symbolic, one as numeric *rules* for Genta's supposed Ferrari:

```
ClearAll[b];
b[symbolic] = {b0, b1 / MN, b2 / K, b3 / MN, b4 / K, b5 / KN,
   b6 / KN^2, b7 / KN, b8, b9 / KN, b10};
b[ferrari] = {b0 → 1.65, b1 → 0., b2 → 1688., b3 → 0.,
   b4 → 229., b5 → 0., b6 → 0., b7 → 0., b8 → -10.,
   b9 → 0., b10 → 0.};
```

So, it's meaningful to do things like this

```
b[symbolic] /. b[ferrari]
```

$$\left\{1.65, \frac{0.}{MN}, \frac{1688.}{K}, \frac{0.}{MN}, \frac{229.}{K}, \frac{0.}{KN}, \frac{0.}{KN^2}, \frac{0.}{KN}, -10., \frac{0.}{KN}, 0.\right\}$$

Define some helper functions and more numeric rules (the nomenclature will be obvious if you have one eye on the equations in part 21; this is a straight transcription):

```
b[n_Integer] := b[symbolic][[n + 1]];
muPeak[fz_] := b[1] fz + b[2]
bigD[fz_] := muPeak[fz] fz
```

```
bb0d[fz_] := (b[3] fz^2 + b[4] fz) E^(-b[5] fz)
bigB[fz_] := bb0d[fz] / (b[0] bigD[fz])
numRulez = {MN → Mega Newton,  KN → Kilo Newton, K → 1000,
   Kilo → 1000,  Mega → 1000000,
   0. → 0, 1. → 1, -1. → 1, 100. → 100};
```

In our numerical exposition of the formula in part 21, we found the constant 0.0822203 popping up over and over. So, test the current forms to see if the same constant appears by substituting the Ferrari's data into them:

```
Simplify[bigB[aa KN] /. b[ferrari]] /. numRulez
0.0822203
```

Add more helpers:

```
bigS[sigma_, fz_] := (100 sigma) + b[9] fz + b10
bigE[fz_] := b[6] fz^2 + b[7] fz + b[8]
```

and, finally, the whole formula:

```
fx[sigma_, fz_] := Module[
  {SB = bigS[sigma, fz] bigB[fz],
   e = bigE[fz]},
  bigD[fz] Sin[b[0] ArcTan[SB + e (ArcTan[SB] - SB)]]]
```

Make a version of the formula with the Ferrari substituted in:

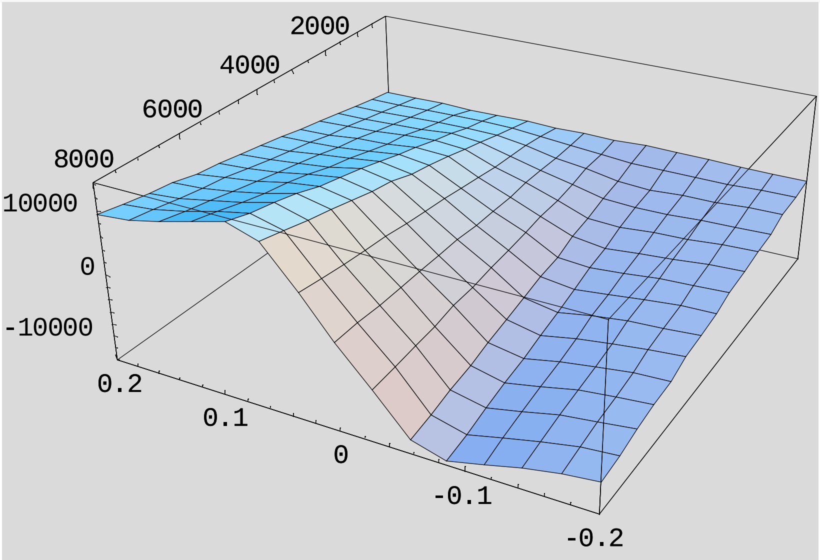
```
fxNum[s_, fz_] := fx[s, fz] / Newton /. b[ferrari] /. numRulez
```

Evaluate this final formula symbolically to check against part 21, then numerically, with a big plot:

```
fxNum[sig, zed]
```

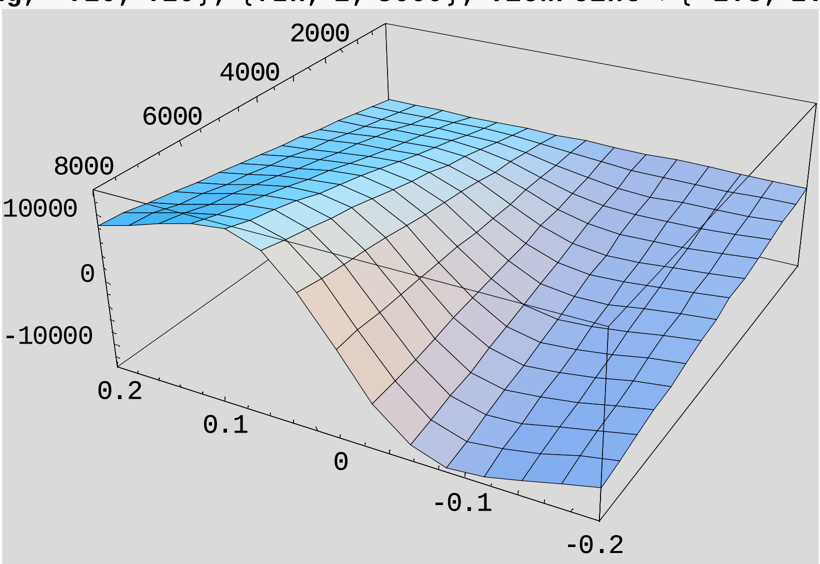
$$\frac{1}{\mathrm{Newton}}(1.688\ \mathrm{zed}\ \mathrm{Sin}[1.65\ \mathrm{ArcTan}[8.22203\ \mathrm{sig} - 10.\ (-8.22203\ \mathrm{sig} + \mathrm{ArcTan}[8.22203\ \mathrm{sig}])]])$$

```
Plot3D[fxNum[sigma, fz Newton],
  {sigma, -.20, .20}, {fz, 1, 8000}, ViewPoint → {-1.3, 2.4, 1.5}];
```



We've successfully reproduced the Magic Formula, and this is the plot we want to emulate. So far, there has been nothing new, just recasting of familiar territory in Mathematica, where we can manipulate the data and formulas. The new function is vastly easier to write down (that's the whole point!), and the plot looks very similar:

```
fx2Num[s_, fzN_, A_, B_, P_] := B fzN s / (1 + Abs[A s] ^ P)
Plot3D[fx2Num[sig, fzn, 9.625, 31, 2.375],
  {sig, -.20, .20}, {fzn, 1, 8000}, ViewPoint → {-1.3, 2.4, 1.5}];
```



It remains only to show how to obtain the parameter values

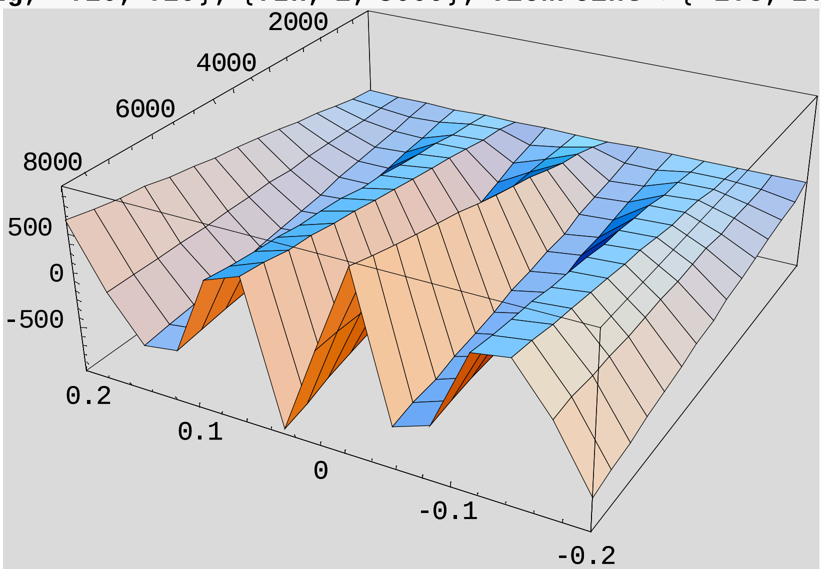
$$(A, B, P) = (9.625, 31.0, 2.375).$$

The short answer is by the ***least squares method***. The slightly longer answer is to subtract the two plots, sum up the squares of the differences, and then find those values of (A, B, P) that minimize this sum of squares. In the following, I use the already known best values, as if they were handed to me by an oracle. The reason I do this is that it keeps the presentation short: you don't have to look at a bunch of trial-and-error plots gone haywire. But imagine that you don't know the values as I talk you through the process of finding them. I talk you through it because the process is generic, meaning that you can apply it to parameter-searching problems similar to this one.

First, plot the errors, that is, the difference between the two forms.

```
errdiff1[sig_, fzn_, A_, B_, P_] :=
 (fxNum[sig, fzn Newton] - fx2Num[sig, fzn, A, B, P])
```

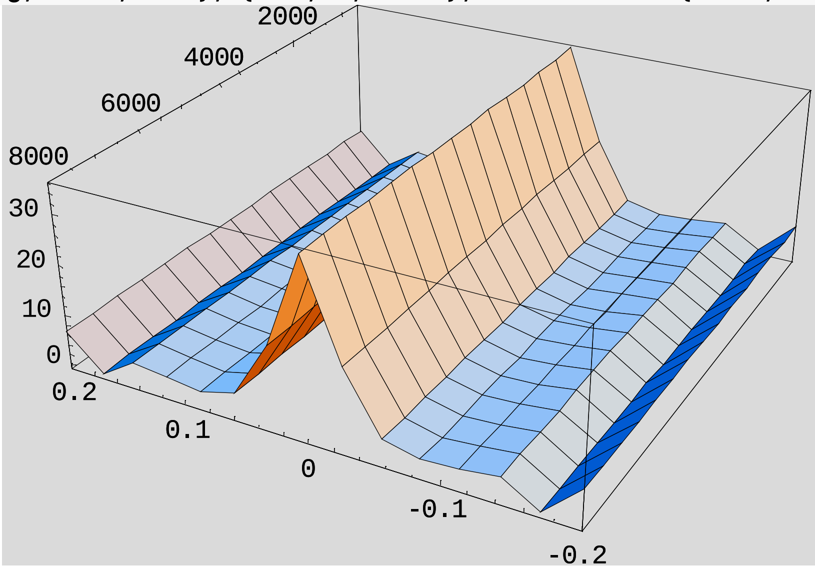
```
Timing[Plot3D[errdiff1[sig, fzn, 9.625, 31, 2.375],
  {sig, -.20, .20}, {fzn, 1, 8000}, ViewPoint → {-1.3, 2.4, 1.5}]]
```




```
{0.812 Second , - SurfaceGraphics -}
```

Note that the maximum magnitude of the errors is about 800 N, or less than 10% of the maximum force. Given that the shape of the plot is very similar to that of Pajecka, hence, that the behavioral characteristics will be similar, we should feel encouraged to go on, with the reservation that the errors are large at the origin, where the force is small. In fact, the errors do reach around 32% at the origin, as shown in the next plots

```
errpct[sig_, fzn_, A_, B_, P_] :=
 100 Abs[(1 - fx2Num[sig, fzn, A, B, P] / fxNum[sig, fzn Newton])]
Timing[Plot3D[errpct[sig, fzn, 9.625, 31, 2.375],
  {sig, -.20, .20}, {fzn, 1, 8000}, ViewPoint -> {-1.3, 2.4, 1.5}]]
```



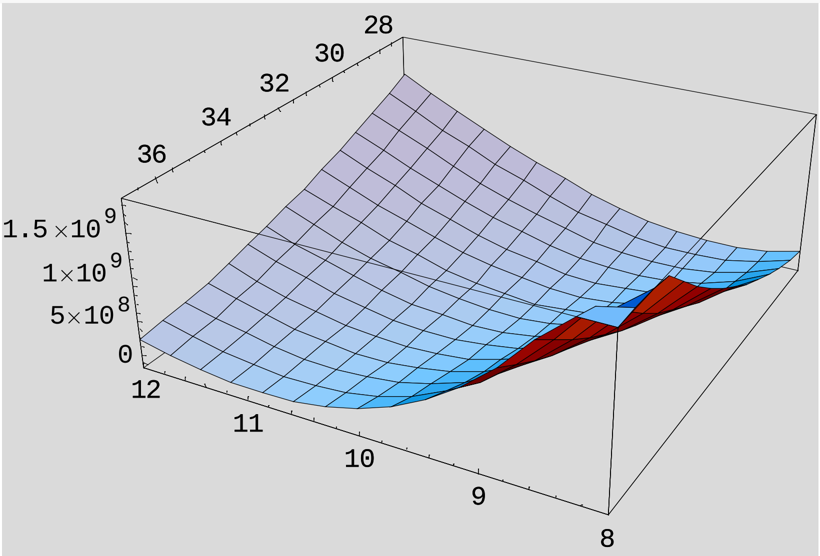
However, this does not concern me very much, for a couple of reasons. First, consider the fact that most authors are satisfied to use a *linear* approximation near the origin. Witness the ever-present *cornering stiffness* in the lateral case. Neither of our forms is particularly linear and would probably deviate from a linear approximation at least as much as they deviate from one another. If a linear approximation to Pajecka is good enough, why wouldn't a handy nonlinear approximation, which happens to have much better overall behavior, be good enough? Secondly, in racing simulation, we are seldom in the linear region, and we should be much more concerned with the differences near the force maxima, around $s = 0.15$. At these values, the errors are much smaller, well under 10%.

Turning attention back to the plot of errors, notice that some errors are negative and some are positive. That's the reason that we square them: we want the overall ***sum*** to be positive for any values of the parameters. We don't want to add negative errors to positive errors and risk getting a low estimate of variance. Note that we might just as well add the absolute values rather than the squares, but squares are much easier to manage symbolically since they are smooth (differentiable) at the origin. Also, squaring errors amounts to an assumption of independence: it treats the errors just like components of a vector in a large-dimensional Euclidean space. Historically, a considerable number of tools were developed to solve such problems analytically (symbolically), so squares it is. In fact, we *could* approach the present problem symbolically with Mathematica, and we might do that in a later installment.

For now, in the interest of getting results quickly, we do a brute-force search for the best values of (A, B, P). I like to call this search method ***archaeological***, because it's the same one archaeologists use to search a site for artifacts: grid off the search space, then exhaustively examine each grid cell. Since (A, B, P) spans a three-dimensional space, each cell is a cube, just like a real archaeological dig! Note that archaeological search is dumb: it's much more clever to do hill climbing (steepest descent), simulated annealing, Kernighan-Lin, or symbolic solution. However, precisely because archaeological search *is* so dumb, it's nearly trivial to code and debug. For many problems, human coding time is more valuable than computer time, and that's the case here. So we let the computer do a bit of extra heavy sweating to save our effort. Note we don't have to do these searches in real time in the simulation: we only have to do one search for each set of Pajecka parameters. If a simulation has 20 different kinds of tires, 20 searches are done ahead of time and the (A, B, P) parameters go into the simulation in a table. If it *were* necessary to search in real time, then cleverer methodology and coding would be justified.

First, zoom in on some two-dimensional slices of the search space and see if we're near a minimum. Look at χ^2, the sum over σ from, say, -0.2 to 0.2 by 0.02 and F_z from, say, 1 to 8000 by 500 of the squares of the errors. Keep the exponent P fixed for this slice and vary A from 8 to 12 and B from 27 to 37.

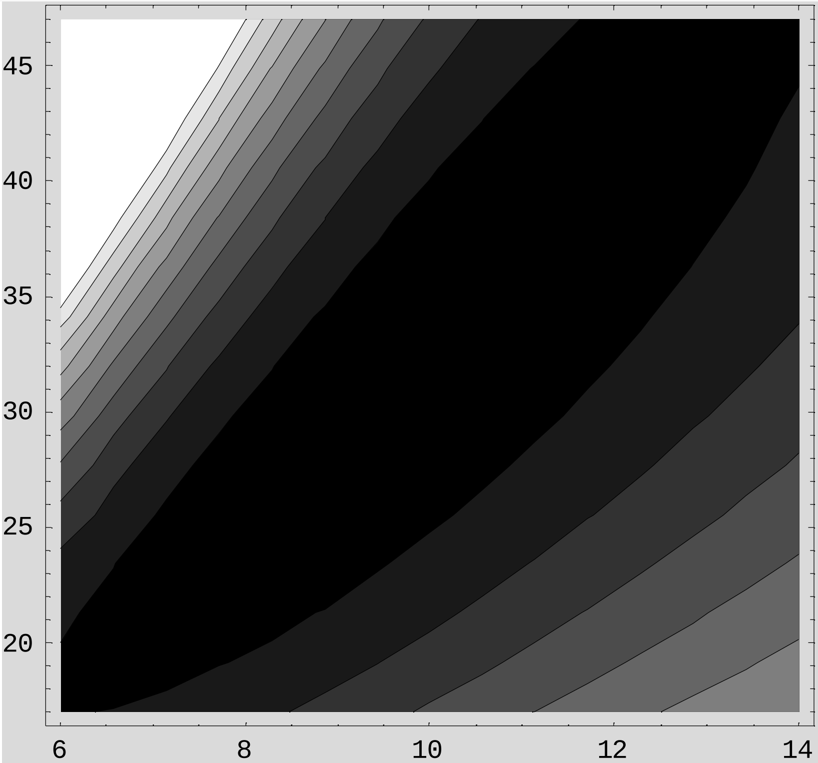
```
s2[A_, B_, P_] := Module[
  {v = 0},
  Do[v += errdiff2[s, f, A, B, P],
   {s, -0.2, 0.2, 0.02}, {f, 1, 8000, 500}];
  v]
Timing[Plot3D[s2[i, j, 2.375],
  {i, 8, 12}, {j, 27, 37}, ViewPoint → {-1.3, 2.4, 1.5}]]
```




```
{247.496 Second , - SurfaceGraphics -}
```

Notice the timing: this plot took some time to produce, so we're having to wait a little for results. More importantly, notice the broad trough at the bottom, suggesting that a minimum can be found in the trough somewhere. A contour plot of the same region gives us even more confidence that we're on the correct archaeological site and can begin digging:

```
Timing[ContourPlot[s2[i, j, 2.375], {i, 6, 14}, {j, 17, 47}]]
```

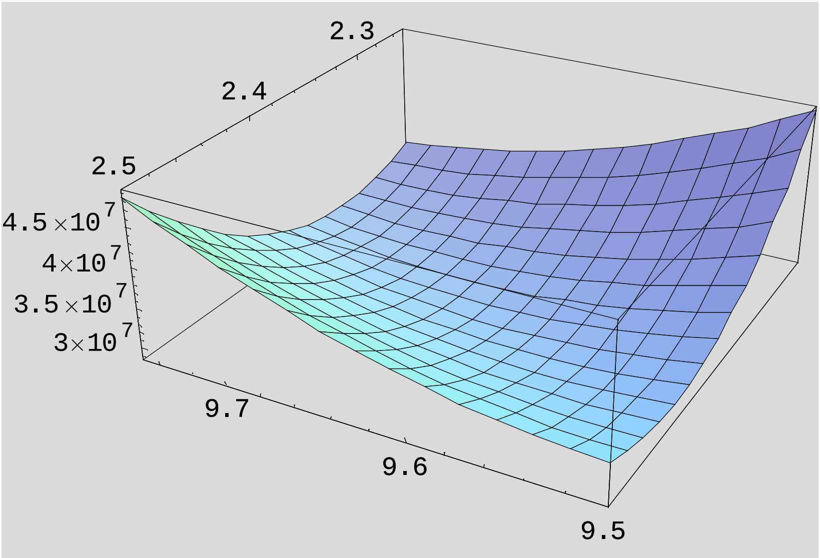



```
{256.328 Second , - ContourGraphics -}
```

The contour plot shows that the trough is roughly elliptical and most likely holds a minimum of χ^2. It is not *certain* to do, and this business of parameter searching can be quite hazardous—it's at least as tricky as integration (part 28). However, for this application, we're almost splitting hairs. The "Magic Trick" function, by eyeball, looks like an effective substitute for full Pajecka, for a wide range of (A, B, P) parameters. We can probably get "close enough" just by eyeball. In fact, the trough in the plot is quite broad, suggesting that a large range of values produce almost equally good results. In any event, we should perform a modicum of due diligence just in the name of professionalism of practice.

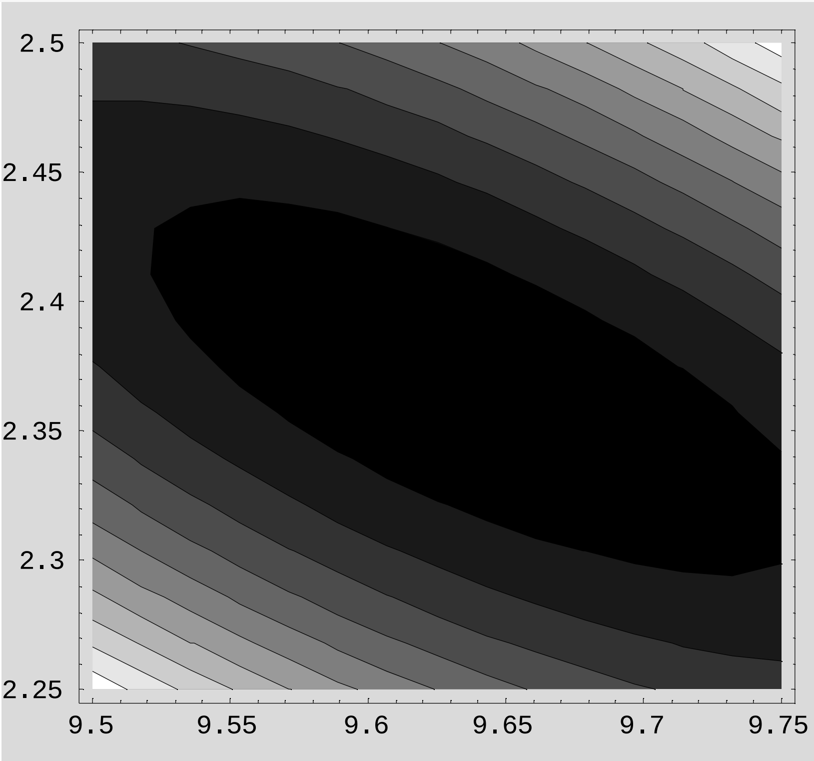
Look at another slice, this time fixing B.

```
Timing[Plot3D[s2[i, 31, k],
  {i, 9.5, 9.75}, {k, 2.25, 2.5}, ViewPoint → {-1.3, 2.4, 1.5}]]
```



{262.708 Second , - SurfaceGraphics -}

```
Timing[ContourPlot[s2[i, 31, k], {i, 9.5, 9.75}, {k, 2.25, 2.5}]]
```



{263.458 Second , - ContourGraphics -}

This slice has almost the same character as the previous one: a broad, narrow trough that likely contains the minimum of χ^2. We are now fortified with confidence and proceed to search:

```
run[step_, {alo_, ahi_}, {blo_, bhi_}, {plo_, phi_}] :=
 Timing[Module[{ax = 0, bx = 0, px = 0,
    ebest = $MaxMachineNumber, etest = 0,
    abest = 0, bbest = 0, pbest = 0},
   Do[
    etest = s2[ax, bx, px];
    If[etest < ebest, ebest = etest;
     abest = ax; bbest = bx; pbest = px;
     Print[{ebest, abest, bbest, pbest}], Null],
    {ax, alo, ahi, step},
    {bx, blo, bhi, step},
    {px, plo, phi, step}];
   {ebest, abest, bbest, pbest}]]
run[1/8., {9.5, 9.75}, {30.50, 31.50}, {2.250, 2.5}]
```

$\{4.08339 \times 10^7, 9.5, 30.5, 2.25\}$

$\{2.90981 \times 10^7, 9.5, 30.5, 2.375\}$

$\{2.90128 \times 10^7, 9.625, 30.875, 2.375\}$

$\{2.89942 \times 10^7, 9.625, 31., 2.375\}$

$\{82.208\ \text{Second}, \{2.89942 \times 10^7, 9.625, 31., 2.375\}\}$

Voila! We have found the promised values $(A, B, P) = (9.625, 31.0, 2.375)$.

Notice how trivial is the searching code: just a triple loop over the parameter space. However, the embedded function, **s2**, is, itself a double loop over the slip s and vertical load F_z, so we have a quintuple loop. I invite you to experiment with longer-running, finer-grained searches by varying the first input to **run**. The inputs are the search step size and the search boundaries for the three parameters. All three parameters are stepped by the same increment. I warn you that as the search grain is made finer, the minimum becomes slippery and the boundaries of the search have to be adjusted. This is the trial-and-error that I mentioned at the beginning of the article. It doesn't matter much that we have an *absolute* minimum in our application. But, in other applications, this slipperiness could be critically important and we would be forced to abandon archaeological search for other methods.